Computing Science Group

FULL ABSTRACTION FOR NOMINAL EXCEPTIONS AND GENERAL REFERENCES

Nikos Tzevelekos
nikt@comlab.ox.ac.uk

CS-RR-07-08

Oxford University Computing Laboratory
Wolfson Building, Parks Road, Oxford, OX1 3QD

Abstract

Game semantics has been used with considerable success in formulating fully abstract semantics for languages with higher-order procedures and a wide range of computational effects. Recently, nominal games have been proposed for modeling functional languages with names and nominal general references. Here we make a step further by developing a fully abstract semantics for a language with nominal exceptions and nominal general references.

1 Introduction

A prevalent feature of programming languages is the use of exceptions for raising and handling eccentric program behavior, and more generally for manipulating the flow of control. It is a key feature, for example, of ML, Java and C++. The raising of an exception forces a program to *escape out of its context and to the nearest available exception-handler.*[1] The effect of overriding nested behavior, while of fundamental importance to the programmer, is very difficult to capture semantically with full-abstraction. The semanticist's task becomes even more daunting if general references are added in the specification, and hence phenomena of dynamic update and interference come into the picture.

The first (and only) fully abstract model of a language with exceptions was presented by Laird in [10]. It formed part of a series of works that, starting in the mid 90's with full-abstraction for PCF (appearing in print somewhat later in [2, 8]), provided fully abstract denotational semantics for programming languages with higher-order procedures and a wide range of computational effects via game semantics. The calculus examined and modeled in [10] contains also general references, and it encodes exceptions as product terms of raise/handle type, and references as read/write types. With this convention, however, and in order to achieve full-abstraction, one needs to include in the language non-exception terms of exception type (*bad exceptions*) and non-reference terms of reference type (*bad variables*). This allows for unwanted behaviors and, amongst other things, prohibits the use of equality tests for references.

In this paper we obtain the *first full-abstraction result for a statically-scoped language with (dynamically bound, locally declared) good exceptions, general references and good variables*, which faithfully reflects the practice, and reaches the expressivity, of real programming languages such as ML. We follow the alternative –nominal– approach of treating exceptions and references separately from variables, as *names*, extending the ν-calculus of Pitts and Stark [16]. The ν-calculus is a paradigmatic λ-calculus with names, in which names are constant terms of ground type that "...are created with local scope, can be tested for equality and can be passed around via function application, but that is all". Here we use two sorts of names: exception-names and reference-names. All names are created locally; exception-names can also be raised and handled, and reference-names

[1] at least as long as *dynamically bound, locally declared* exceptions are concerned, which are the type of exceptions we examine in this paper.

can also be tested for equality, passed around, dereferenced and updated. We call the resulting calculus $\nu\varepsilon\rho$.

Nominal games were introduced in [1] as the basis for the first fully abstract model of the ν-calculus,[2] and were further developed in [17] to provide a fully abstract semantics for $\nu\rho$, the calculus extending ν by use of names for general references. They constitute a version of Honda-Yoshida CBV-games [7] with local state [14], built inside the universe of nominal sets [6, 15]. On the other hand, a major contribution in the study of semantics for languages with computational effects was the introduction of *monads* [13] as a generic tool for effect-modeling. The passage from the model of the ν-calculus to that of $\nu\rho$ was based on the use of a *store-monad*. The next logical step would be to examine whether the use of an additional *exception-monad* could yield a model of $\nu\varepsilon\rho$, and this step we take here. The use of an exception-monad involves the introduction of exceptions without use of exception types: any computation type includes an exception by means of the monad, but in order for a term to be raised as an exception it must evaluate to an exception-name.

We have deviated from the model of [17] also in another aspect. In that work a model was constructed as a family of categories, one category for each local state in the language. Here, instead, we present everything in a single category and use *comonads to model local state*. This description not only provides a succinct and intuitive description of what constitutes an *abstract model* of $\nu\varepsilon\rho$, but also makes the construction of the actual model in nominal games easier.

Summarising, the contributions of this paper are: a) the introduction of a λ-calculus with nominal (good) exceptions and nominal general (good) references; b) the description of abstract categorical models of the language in a monadic-comonadic setting; c) the construction of a fully abstract model using nominal games. We think that (b) should be further examined, with the objective of fully understanding the abstract categorical structure of nominal languages, and of cleanly describing the nominal feature as a computational effect.

2 Theory of Nominal Sets

We give a short overview of nominal sets, which will be used as the basis for all constructions presented in this paper. Intuitively, nominal sets are sets whose elements are built over a finite number of *names*, and which are acted upon by finite name-permutations.

Assume a countably infinite set TY of types denoted by A, B, C and variants, and for each type A assume a countably infinite set of names $\mathbb{A}_A$. Moreover, assume another set of names $\mathbb{A}_E$. The elements of $\mathbb{A}_A$ are ***reference-names*** to type A, while the elements of $\mathbb{A}_E$ are ***exception-names***. We let

$$\mathbb{A} \triangleq \mathbb{A}_E \uplus \bigcup_{A \in \mathrm{TY}} \mathbb{A}_A$$

[2] A different version of nominal games was introduced in [11].

be the set of (general) names. Names will be generally denoted by a, b, c and variants; reference-names will be specifically denoted by $\ddot{a}, \ddot{b}, \ddot{c}$ and variants, and exception-names will be denoted by $\dot{a}, \dot{b}, \dot{c}$ and variants. We write $\mathrm{PERM}(\mathbb{A}_A)$ for the group of finite permutations of $\mathbb{A}_A$, and similarly for $\mathrm{PERM}(\mathbb{A}_E)$. We take $\mathrm{PERM}(\mathbb{A})$ to be the direct sum of the $\mathrm{PERM}(\mathbb{A}_X)$'s,

$$\mathrm{PERM}(\mathbb{A}) \triangleq \bigoplus_{X \in \{E\} \uplus \mathrm{TY}} \mathrm{PERM}(\mathbb{A}_X)$$

that is elements of $\mathrm{PERM}(\mathbb{A})$ are those permutations of $\mathbb{A}$ that can be described as finite sequences of permutations from the $\mathrm{PERM}(\mathbb{A}_X)$'s. $(a\ b)$ denotes the permutation that only swaps names a and b (of same type) and $\mathtt{id}$ denotes the identity permutation; permutations in general are denoted by π and variants.

A ***nominal set*** X is a set equipped with an action from $\mathrm{PERM}(\mathbb{A})$, that is a function $\_ \circ \_ : \mathrm{PERM}(\mathbb{A}) \times X \to X$ such that, for any $\pi, \pi' \in \mathrm{PERM}(\mathbb{A})$ and $x \in X$,

$$\pi \circ (\pi' \circ x) = (\pi \circ \pi') \circ x \qquad \mathtt{id} \circ x = x$$

Moreover, all $x \in X$ have ***finite support*** $\mathtt{S}(x) \subseteq \mathbb{A}$, which is the least set $S \subseteq \mathbb{A}$ satisfying:

$$\forall \pi \in \mathrm{PERM}(\mathbb{A}).\ (\forall a \in S.\ \pi(a) = a) \implies \pi \circ x = x$$

For $x \in X$ and $a \in \mathbb{A}$ we say that a ***is fresh for*** x, and write $a \# x$, if $a \notin \mathtt{S}(x)$.

We can see that $\mathbb{A}$ in particular is a nominal set with each name a having support $\{a\}$. If Y is a nominal set and $X \subseteq Y$ then X is a ***nominal subset*** of Y if X is closed under permutations, these acting as on Y. If $X \subseteq Y$ is a nominal subset then so is $\bar{X} \triangleq Y \setminus X$. If X, Y are nominal sets then their cartesian product $X \times Y$ is also a nominal set, with permutations defined componentwise. Similarly, $\mathbb{A}^{\#}$, the set of ***finite lists of distinct names***, is a nominal set; we denote its elements by $\vec{a}, \vec{b}, \vec{c}$ and variants. Moreover, a relation $R \subseteq X \times Y$ is a ***nominal relation*** if it is a nominal subset of $X \times Y$. Concretely, R is nominal iff, for any permutation π and $(x, y) \in X \times Y$, $x\, R\, y \iff (\pi \circ x)\, R\, (\pi \circ y)$. Accordingly, $f : X \to Y$ is a ***nominal function*** if $f(\pi \circ x) = \pi \circ f(x)$, for any $x \in X$ and π. For example, cartesian product projections are nominal functions.

In nominal sets we can succinctly define ***name-abstraction***: for each $a \in \mathbb{A}$ and $x \in X$ let

$$\langle a \rangle x \triangleq \{(b, y) \in \mathbb{A} \times X \mid (b = a \vee b \# x) \wedge y = (a\ b) \circ x\}$$

We can show $\mathtt{S}(\langle a \rangle x) = \mathtt{S}(x) \setminus \{a\}$. Another form of abstraction involves ***equivariancia-tion***, that is abstracting the whole support of an element by forming its orbit under all permutations: for any $x \in X$ let

$$[x] \triangleq \{\pi \circ x \mid \pi \in \mathrm{PERM}(\mathbb{A})\}$$

Clearly, $\mathtt{S}([x]) = \varnothing$. For example, for names $\ddot{a} \in \mathbb{A}_A$ and $\dot{a} \in \mathbb{A}_E$, $[\ddot{a}\dot{a}]$ is the set of all lists of names comprising precisely a reference-name of type A followed by an exception-name, that is,

$$[\ddot{a}\dot{a}] = \{\ddot{b}\dot{b} \mid \ddot{b} \in \mathbb{A}_A \wedge \dot{b} \in \mathbb{A}_E\}$$

Alternatively, $[\ddot{a}\dot{a}]$ can be seen as a *generic list* of a reference-name of type A followed by an exception-name.

The notion of support can be strengthened as follows. We say that an element x of a nominal set X has ***strong support*** if

$$\forall \pi \in \mathrm{PERM}(\mathbb{A}).\, (\forall a \in \mathtt{S}(x).\, \pi(a) = a) \iff \pi \circ x = x$$

We say X is a ***strong nominal set*** if all its elements have strong support. We can see that all constructions of the two previous paragraphs are inherited by strong nominal sets. We let $\mathbf{sNom}_{\mathrm{TYE}}$ be the category of strong nominal sets (on $\mathbb{A}$) and nominal functions.

3 $\nu\varepsilon\rho$-calculus

The $\nu\varepsilon\rho$-calculus extends the ν-calculus of Pitts and Stark [16] by using names for general references and exceptions.

Definition 1 The $\nu\varepsilon\rho$-calculus is a functional calculus of nominal references and exceptions. Its types are given as follows.

$$\mathrm{TY} \ni A, B ::= \mathbb{1} \mid \mathbb{N} \mid [A] \mid A \to B \mid A \otimes B$$

Terms of type $\mathbb{1}$ are commands, type $\mathbb{N}$ is for natural numbers, type $[A]$ for references to type A, and the rest are arrow and product types. Terms are given as elements of $\mathbf{sNom}_{\mathrm{TYE}}$, as follows.

$\mathrm{TE} \ni M, N ::= x \mid \lambda x.M \mid M\, N$	λ-calculus
$\mid \mathtt{skip}$	return
$\mid n \mid \mathtt{pred}\, M \mid \mathtt{succ}\, N$	arithmetic
$\mid \mathtt{if0}\ M\ \mathtt{then}\ N_1\ \mathtt{else}\ N_2$	if_then_else
$\mid \langle M, N\rangle \mid \mathtt{fst}\, M \mid \mathtt{snd}\, N$	pair/ projections
$\mid \ddot{a} \mid [M = N]$	reference/ ref. equality
$\mid M := N \mid !M$	update/ dereferencing
$\mid \mathtt{raise}\, \dot{a}$	raise exception
$\mid \mathtt{try}\ M\ \mathtt{handle}\ \dot{a} \Rightarrow N$	try/handle exception
$\mid \nu a.M$	ν-abstraction

Each name $\ddot{a}$ is taken from $\biguplus_{A \in \mathrm{TY}} \mathbb{A}_A$, and each name $\dot{a}$ is taken from $\mathbb{A}_E$.

Of the terms above, the values are:

$$\mathrm{VA} \ni V, W ::= n \mid \mathtt{skip} \mid \ddot{a} \mid x \mid \lambda x.M \mid \langle V, W\rangle$$

The typing system involves terms in environments $\vec{a} \mid \Gamma$, where $\vec{a}$ a list of (distinct) names and Γ a finite set of variable-type pairs. Some of its rules are the following.

$$\frac{}{\vec{a} \mid \Gamma \vdash \mathtt{raise}\, \dot{a} : A}\ \dot{a} \in \mathtt{S}(\vec{a}) \qquad \frac{}{\vec{a} \mid \Gamma \vdash \ddot{a} : [A]}\ \ddot{a} \in \mathtt{S}(\vec{a}) \cap \mathbb{A}_A$$

$$\frac{\vec{a}a \mid \Gamma \vdash M : B}{\vec{a} \mid \Gamma \vdash \nu a.M : B} \qquad \frac{\vec{a} \mid \Gamma \vdash M : [A] \qquad \vec{a} \mid \Gamma \vdash N : [A]}{\vec{a} \mid \Gamma \vdash [M = N] : \mathbb{N}}$$

$$\frac{\vec{a} \mid \Gamma \vdash M : [A]}{\vec{a} \mid \Gamma \vdash !M : A} \qquad \frac{\vec{a} \mid \Gamma \vdash M : [A] \qquad \vec{a} \mid \Gamma \vdash N : A}{\vec{a} \mid \Gamma \vdash M := N : \mathbb{1}}$$

$$\frac{\vec{a} \mid \Gamma \vdash M : A \qquad \vec{a} \mid \Gamma \vdash N : A}{\vec{a} \mid \Gamma \vdash \mathtt{try}\ M\ \mathtt{handle}\ \dot{a} \Rightarrow N : A}\ \dot{a} \in \mathtt{S}(\vec{a})$$

▲

Permutations act on TE componentwise, that is, for any $\pi \in \mathrm{PERM}(\mathbb{A})$,

$$\pi \circ a = \pi(a) \quad \pi \circ \nu a.M = \nu(\pi \circ a).(\pi \circ M) \quad \pi \circ x = x \quad \pi \circ \lambda x.M = \lambda(\pi \circ x).(\pi \circ M) \quad \textit{etc.}$$

Note that there are two types of binding in the syntax, variable-binding and name-binding, and each of these yields its own notion of α-equivalence. The set of ***free variables*** of a term is defined as,

$$\mathtt{fv}(x) \triangleq \{x\} \quad \mathtt{fv}(\lambda x.M) \triangleq \mathtt{fv}(M) \setminus \{x\} \quad \mathtt{fv}(n) = \mathtt{fv}(a) \triangleq \varnothing$$

plus standard rules for the other non-binding constructs. A term M is ***closed*** if $\mathtt{fv}(M)$ is empty. Similarly, the set of ***free names*** of a term is defined as,

$$\mathtt{fn}(a) \triangleq \{a\} \quad \mathtt{fn}(\nu a.M) \triangleq \mathtt{fn}(M) \setminus \{a\} \quad \mathtt{fn}(n) = \mathtt{fn}(x) \triangleq \varnothing$$

plus standard rules for the other non-binding constructs. α-equivalence for variable-binding, henceforth called α_V-equivalence and written $=_{\alpha_V}$, is defined as usually.[3] α-equivalence for name-binding, henceforth called α_N-equivalence and written $=_{\alpha_N}$, is defined by recursion (on term size) as follows,

$$\frac{}{M =_{\alpha_N} M}\ M = x, a, n \qquad \frac{\text{for cofinitely many } b.\ (a\ b) \circ M =_{\alpha_N} (a'\ b) \circ M'}{\nu a.M =_{\alpha_N} \nu a'.M'} \qquad \frac{M =_{\alpha_N} M'}{\lambda x.M =_{\alpha_N} \lambda x.M'}$$

[3] Since the formality of definitions of α_V-equivalence available out there varies considerably (with occasional occurrences of incorrect definitions) perhaps it is worthwhile providing a quick definition, taken from [9].

First we define angle-brackets substitution to be the simple (i.e. not capture-avoiding) var-for-var substitution with $(\lambda x.M) < y/x > \triangleq \lambda x.M$. Then we define $=_{\alpha_V}$ by recursion (on term size) as:

$$\frac{}{M =_{\alpha_V} M}\ M = x, a, n \qquad \frac{\text{for cofinitely many } y.\ M < y/x > =_{\alpha_V} M' < y/x' >}{\lambda x.M =_{\alpha_V} \lambda x'.M'} \qquad \frac{M =_{\alpha_V} M'}{\nu a.M =_{\alpha_V} \nu a.M'}$$

plus standard rules for the other non-binding constructs. Note the similarity with the definition of $=_{\alpha_N}$.

plus standard rules for the other non-binding constructs. The definition is taken from [6], and it captures the usual notion of α-equivalence, i.e. it equates terms up to choice of bound names (v. [6, proposition 2.2]).

The casting of our calculus in nominal sets equips us with a well-behaved action of name-permutation on terms. We trivially have that $a, b \# M \implies (a\ b) \circ M = M$, any term M and $a, b \in \mathbb{A}$. Moreover, the following hold.

Proposition 2 *For all terms M, N and $a, b \in \mathbb{A}$,*

- $M =_{\alpha_N} N \implies (a\ b) \circ M =_{\alpha_N} (a\ b) \circ N$,
- $a, b \notin \mathtt{fn}(M) \implies (a\ b) \circ M =_{\alpha_N} M$.

Proof: By induction on M. ■

The above proposition implies that the second rule for $=_{\alpha_N}$ reduces to $\dfrac{M =_{\alpha_N} M'}{\nu a.M =_{\alpha_N} \nu a.M'}$ for $a = a'$. Now, we take the usual step of equating terms up to α-equivalence.

> We assume the set of terms is quotiented by α-equivalence for both binding mechanisms, that is we equate terms up to choice of bound variables and bound names.

We proceed in defining reduction in $\nu\varepsilon\rho$, which is call-by-value. The reduction calculus is defined in exceptional store environments, S, which may also enlist exception-names.

Definition 3 We define exceptional store environments by:

$$S ::= \epsilon \mid a, S \mid \ddot{a} :: V, S$$

For each S we define its domain to be the list of names enlisted in S. We only consider environments whose domains are lists of distinct names, and write $S \vDash_{\Gamma,A} M$, or simply

$S \models M$, only if $\mathtt{dom}(S) \mid \Gamma \vdash M : A$ is derivable. Reduction rules are as below,

$$\text{LAM}\ \frac{}{S \models !\ddot{a} \longrightarrow S, \ddot{a} :: V, S' \models V}$$

$$\text{DRF}\ \frac{}{S, \ddot{a} :: V, S' \models !\ddot{a} \longrightarrow S, \ddot{a} :: V, S' \models V}$$

$$\text{UPD}\ \frac{}{S, \ddot{a}(:: W), S' \models \ddot{a} := V \longrightarrow S, \ddot{a} :: V, S' \models \mathtt{skip}}$$

$$\text{CHK}\ \frac{}{S \models [\ddot{a} = \ddot{b}] \longrightarrow S \models n}\ \begin{array}{l} n{=}1 \text{ if } \ddot{a} \# \ddot{b} \\ n{=}0 \text{ if } \ddot{a} {=} \ddot{b} \end{array}$$

$$\text{NEW}\ \frac{}{S \models \nu a.M \longrightarrow S, b \models (a\ b) \circ M}\ (b \# S)$$

$$\text{HL}\ \frac{}{S \models \mathtt{try}\ (\mathtt{raise}\ \dot{a})\ \mathtt{handle}\ \dot{a} \Rightarrow N \longrightarrow S \models N}$$

$$\text{VHL}\ \frac{}{S \models \mathtt{try}\ V\ \mathtt{handle}\ \dot{a} \Rightarrow N \longrightarrow S \models V}$$

$$\text{NHL}\ \frac{}{S \models \mathtt{try}\ (\mathtt{raise}\ \dot{b})\ \mathtt{handle}\ \dot{a} \Rightarrow N \longrightarrow S \models \mathtt{raise}\ \dot{b}}\ \dot{a} \# \dot{b}$$

$$\text{XPN}\ \frac{}{S \models \mathrm{Z}[\mathtt{raise}\ \dot{a}] \longrightarrow S \models \mathtt{raise}\ \dot{a}}$$

$$\text{CTX}\ \frac{S \models M \longrightarrow S' \models M'}{S \models \mathrm{E}[M] \longrightarrow S' \models \mathrm{E}[M']}$$

plus standard CBV rules for `fst`, `snd`, `if0`, `pred` and `succ`. Unhandled evaluation contexts Z[_] are of the forms:

$$[\_ = N]\ ,\ [\ddot{a} = \_]\ ,\ !\_\ ,\ \_ := N\ ,\ \ddot{a} := \_\ ,$$
$$(\lambda x.N)\ \_\ ,\ \_\ N\ ,\ \mathtt{if0}\ \_\ \mathtt{then}\ N\ \mathtt{else}\ N',$$
$$\mathtt{fst}\ \_\ ,\ \mathtt{snd}\ \_\ ,\ \mathtt{pred}\ \_\ ,\ \mathtt{succ}\ \_\ ,\ \langle \_, N \rangle\ ,\ \langle V, \_ \rangle$$

and (general) evaluation contexts E[_] are of the forms:

$$\mathrm{Z}[\_]\ ,\ \mathtt{try}\ \_\ \mathtt{handle}\ \dot{a} \Rightarrow N$$

▲

Observe the fact that *exceptions carry only their names*. This, although very intuitive, could be seen as a restriction in the language, as usually one wants exceptions to carry some secondary data. Nevertheless, in the presence of local references such data can be easily carried through the store, so we don't think there is any expressivity loss here.

We take *observable terms* to be the constants of type $\mathbb{N}$, and around them we build the notion of observational equivalence.

Definition 4 For typed terms $\vec{a} \mid \Gamma \vdash M : A$ and $\vec{a} \mid \Gamma \vdash N : A$, define $\vec{a} \mid \Gamma \vdash M \lessapprox N$ to be the assertion:

$$\begin{array}{l} \text{for any variable- and name-closing context } \mathrm{C}[\_] : \mathbb{N}, \\ \exists S'.(\vDash \mathrm{C}[M] \longrightarrow\!\!\!\!\!\rightarrow\ S' \vDash 0) \implies \exists S''.(\vDash \mathrm{C}[N] \longrightarrow\!\!\!\!\!\rightarrow\ S'' \vDash 0) \end{array}$$

We usually omit $\vec{a}$ and Γ and write simply $M \lessapprox N$. ▲

3.1 Semantics

We examine sufficient conditions for a fully-abstract semantics of $\nu\varepsilon\rho$, specifying thus ***$\lambda_{\nu\varepsilon\rho}$-models***. Translating each term M into an object $[\![M]\!]$ of a $\lambda_{\nu\varepsilon\rho}$-model $\mathcal{M}$ and assuming a preorder $\lesssim$ in $\mathcal{M}$, full-abstraction amounts to the assertion:

$$M \lessapprox N \iff [\![M]\!] \lesssim [\![N]\!] \tag{FA}$$

3.1.1 Monads and Comonads

The semantics we use is a *monadic* and *comonadic* one, over a computational monad T (v. [13]) and a family of local-state comonads $Q = \langle Q^{\vec{a}} \rangle_{\vec{a} \in \mathbb{A}^\#}$ (v. [4]), so that the morphism related to each typed term $\vec{a} \mid \Gamma \vdash M : A$ is of the form $[\![M]\!] : Q^{\vec{a}}[\![\Gamma]\!] \longrightarrow T[\![A]\!]$.

Recall that a strong monad over a category $\mathcal{C}$ with binary products is a quadruple (T, η, μ, τ) where:

- $T : \mathcal{C} \longrightarrow \mathcal{C}$ is an endofunctor,
- η, μ, τ are natural transformations,

$$\begin{array}{rll} \eta_A : A \longrightarrow TA & & \textit{(unit)} \\ \mu_A : T^2A \longrightarrow TA & & \textit{(composition)} \\ \tau_{A,B} : A \times TB \longrightarrow T(A \times B) & & \textit{(strength)} \end{array}$$

  satisfying certain coherence conditions (v. [13]). We write

$$\tau' : T\_ \times \_ \longrightarrow T(\_ \times \_)$$

  for the strength transformation derived from τ and product symmetries, and take

$$\begin{array}{l} \psi_{A,B} : TA \times TB \longrightarrow T(A \times B) \triangleq \tau' ; T\tau ; \mu \\ \psi'_{A,B} : TA \times TB \longrightarrow T(A \times B) \triangleq \tau ; T\tau' ; \mu \end{array}$$

A comonad over a category $\mathcal{C}$ is a triple (T, ε, δ) where:

- $T : \mathcal{C} \longrightarrow \mathcal{C}$ is an endofunctor,

- ε, δ are natural transformations,

$$\begin{array}{ll} \varepsilon_A : TA \longrightarrow A & \textit{(counit)} \\ \delta_A : TA \longrightarrow T^2A & \textit{(duplication)} \end{array}$$

  satisfying the (dualised) monadic conditions (with strengths excluded). In case $\mathcal{C}$ has binary products, we can define a natural transformation $\bar{\zeta}$

$$\bar{\zeta}_{A,B} : T(A \times B) \xrightarrow{\langle T\pi_1, T\pi_2 \rangle} TA \times TB \xrightarrow{\varepsilon_A \times \mathtt{id}_B} A \times TB$$

  so that $(T, \varepsilon, \delta, \bar{\zeta})$ satisfies the strong monadic conditions.

Stronger comonads are obtained by stipulating a transformation ζ on the other direction, as in the case of *strong commonads* (v. [5]). In our case, we stipulate even stronger conditions.

Definition 5 (Product comonad) A comonad (T, ε, δ) with transformation $\bar{\zeta}$ defined as above is called a *product comonad* if $\bar{\zeta}$ is a natural isomorphism. ▲

We write $\zeta : \_ \times T(\_) \longrightarrow T(\_ \times \_)$ for the inverse of $\bar{\zeta}$. Moreover, and as in the case of monadic strengths, we let $\zeta', \bar{\zeta}'$ be their symmetric counterparts. Note that a product comonad T over a category with finite products can be written as

$$T\_ \cong T1 \times \_$$

hence the name. We say that $T1$ is the ***basis of the comonad***.

3.1.2 Precompound monads

Computation in $\nu\varepsilon\rho$ is exception-raising, store-update and fresh-name creation. Hence, our computational monad T can be described as a two-component monad containing a store- and fresh-name-monad on top of an exception-monad. Instead of giving T explicitly as the composition of two monads we will stipulate that T satisfies certain properties of compound monads. These properties define precompound monads and do suffice for describing abstractly the two-component structure of our monad.

Definition 6 (Precompound monad) A strong monad (T, η, μ, τ) is precompound if there exists a natural transformation $\theta : T \to T^2$ such that, for each object A,

- $\theta_A \,;\mu_A = \mathtt{id}_{TA}$
- $\theta_{TA} \,;T\mu_A \,;T\theta_A \,;\mu_{TA} = \mu_A \,;\theta_A$

Moreover, each η_A is an inner- and outer-component arrow, where an arrow $f : A \longrightarrow TB$ is said to be

- an *inner-component arrow* if $f \,;\theta_B = f \,;\eta_{TB}$

- an *outer-component arrow* if $f\,;\theta_B = f\,;T\eta_B$

We write T as (T,η,μ,τ,θ). ▲

So θ is in essence separating the two components in T, so that the morphism $\theta_A : TA \longrightarrow T^2A$ sends the outer T-component of TA to the outer T of T^2A, and the inner T-component of TA to the inner T of T^2A. From this viewpoint, inner-component arrows can be seen as involving computation in the inner-component of T, and similarly for outer-component arrows.

3.1.3 $\lambda_{\nu\varepsilon\rho}$-models

We proceed in introducing abstract models for $\nu\varepsilon\rho$. Note first some notation for name-lists: for name-lists $\vec{a},\vec{b}$ we write

- $\vec{a} \leq \vec{b}$ when $\vec{a}$ is a prefix of $\vec{b}$,
- $\vec{a} \preceq \vec{b}$ when $\vec{a}$ is a (not necessarily initial) sublist of $\vec{b}$.

Definition 7 A $\lambda_{\nu\varepsilon\rho}$-model $\mathcal{M}$ is a triple $\langle \mathcal{M}, T, Q\rangle$ where,

I. $\mathcal{M}$ is a category with finite products, with 1 being the terminal object and $A \times B$ the product of A and B.

II. T is a precompound monad (T,η,μ,τ,θ) and forms a λ_c-model over $\mathcal{M}$ (v. [13]). The T-exponential TB^A is denoted by $A \stackrel{\cdot}{\Rightarrow} TB$, T-currying by Λ^T, and T-evaluation by ev^T.

III. $\mathcal{M}$ contains a natural numbers object $\mathbb{N}$ equipped with successor/predecessor arrows and $n : 1 \longrightarrow \mathbb{N}$, each $n \in \mathbb{N}$.

IV. $\mathcal{M}$ contains, for each $A \in \mathrm{TY}$, an A-names object $\mathbb{A}_A$ and a symmetric name-equality arrow $\mathsf{eq}_A : \mathbb{A}_A \times \mathbb{A}_A \longrightarrow \mathbb{N}$.

V. Q is a family of product comonads $\langle Q^{\vec{a}}, \varepsilon, \delta, \zeta\rangle_{\vec{a}\in\mathbb{A}^\#}$ on $\mathcal{M}$ such that,

(a) $Q^{\vec{a}} = Q^{\vec{a}'}$ if $[\vec{a}] = [\vec{a}']$, $Q^{\epsilon} \cong Id_{\mathcal{M}}$, and $Q^{\ddot{a}} \cong \mathbb{A}_A \times \_$ if $\ddot{a} \in \mathbb{A}_A$,

(b) for any $\vec{a}' \preceq \vec{a}$ there is a comonad morphism $\frac{\vec{a}}{\vec{a}'} : Q^{\vec{a}} \longrightarrow Q^{\vec{a}'}$ such that $\frac{\vec{a}}{\epsilon} = \varepsilon$ and, whenever $\vec{a}' \preceq \vec{a}'' \preceq \vec{a}$,

$$\frac{\vec{a}}{\vec{a}''}\,;\frac{\vec{a}''}{\vec{a}'} = \frac{\vec{a}}{\vec{a}'}$$

(c) for each type A and $\ddot{a},\ddot{b} \in \mathbb{A}_A$ the following commute,

$$\begin{array}{ccc} \mathbb{A}_A & \xrightarrow{\Delta} & \mathbb{A}_A \times \mathbb{A}_A \\ {\scriptstyle !}\downarrow & & \downarrow{\scriptstyle \mathsf{eq}_A} \\ 1 & \xrightarrow[0]{} & \mathbb{N} \end{array} \qquad \begin{array}{ccc} Q^{\ddot{a}\ddot{b}}1 & \xrightarrow{\langle \frac{\ddot{a}\ddot{b}}{\ddot{a}}, \frac{\ddot{a}\ddot{b}}{\ddot{b}}\rangle} & \mathbb{A}_A \times \mathbb{A}_A \\ {\scriptstyle !}\downarrow & & \downarrow{\scriptstyle \mathsf{eq}_A} \\ 1 & \xrightarrow[1]{} & \mathbb{N} \end{array} \qquad \text{(N1)}$$

(d) there exists a natural transformation $\mathtt{new}^{\vec{a}a} : Q^{\vec{a}} \longrightarrow TQ^{\vec{a}a}$ such that, for each $B \in Ob(\mathcal{M})$, $\mathtt{new}^{\vec{a}a}_B$ is an outer-component arrow and the following diagrams commute.

$$\begin{array}{ccc} Q^{\vec{a}}B & \xrightarrow{\langle \mathtt{id}, \mathtt{new}_B \rangle} & Q^{\vec{a}}B \times TQ^{\vec{a}a}B \\ {\scriptstyle \mathtt{new}_B}\downarrow & & \downarrow{\scriptstyle \tau} \\ TQ^{\vec{a}a}B & \xrightarrow[T\langle \frac{\vec{a}a}{\vec{a}}, \mathtt{id} \rangle]{} & T(Q^{\vec{a}}B \times Q^{\vec{a}a}B) \end{array} \qquad \begin{array}{ccc} A \times Q^{\vec{a}}B & \xrightarrow{\mathtt{id} \times \mathtt{new}_B} & A \times TQ^{\vec{a}a}B \\ {\scriptstyle \zeta}\downarrow & & \downarrow{\scriptstyle \tau ; T\zeta} \\ Q^{\vec{a}}(A \times B) & \xrightarrow[\mathtt{new}_{A \times B}]{} & TQ^{\vec{a}a}(A \times B) \end{array} \tag{N2}$$

VI. Taking

$$[\![\mathbb{1}]\!] \triangleq 1\,,\ [\![\mathbb{N}]\!] \triangleq \mathbb{N}\,,\ [\![[A]]\!] \triangleq \mathbb{A}_A\,,\ [\![A \times B]\!] \triangleq [\![A]\!] \times [\![B]\!]\,,\ [\![A \to B]\!] \triangleq [\![A]\!] \tilde{\Rightarrow} T[\![B]\!]$$

$\mathcal{M}$ contains, for each $A \in \mathrm{TY}$, outer-component arrows

$$\mathtt{drf}_A : \mathbb{A}_A \longrightarrow T[\![A]\!] \quad \text{and} \quad \mathtt{upd}_A : \mathbb{A}_A \times [\![A]\!] \longrightarrow T1$$

such that the following diagrams (which describe the specifications for *dereferencing* and *update*) commute,

$$\begin{array}{ccc} \mathbb{A}_A \times [\![A]\!] & \xrightarrow{\langle \pi_1, \mathtt{upd}_A \rangle ; \tau} & T\mathbb{A}_A \\ & {\scriptstyle \langle \pi_2, \mathtt{upd}_A \rangle ; \tau}\searrow & \downarrow{\scriptstyle T\mathtt{drf}_A ; \mu} \\ & & T[\![A]\!] \end{array} \tag{NR}$$

$$\begin{array}{ccc} \mathbb{A}_A \times [\![A]\!] \times [\![A]\!] & & \mathbb{A}_A \times [\![A]\!] \\ {\scriptstyle \Delta \times \mathtt{id} ; \cong}\downarrow & \nearrow{\scriptstyle \pi_2} & \downarrow{\scriptstyle \mathtt{upd}_A} \\ \mathbb{A}_A \times [\![A]\!] \times \mathbb{A}_A \times [\![A]\!] & \xrightarrow[\mathtt{upd}_A \times \mathtt{upd}_A ; \psi]{} & T1 \end{array} \qquad \begin{array}{ccc} Q^{\ddot{a}\ddot{b}}1 \times [\![A]\!] \times [\![B]\!] & & \mathbb{A}_B \times [\![B]\!] \times \mathbb{A}_A \times [\![A]\!] \\ {\scriptstyle \langle \frac{\ddot{a}\ddot{b}}{\ddot{a}}, \frac{\ddot{a}\ddot{b}}{\ddot{b}} \rangle \times \mathtt{id} ; \cong}\downarrow & \nearrow{\scriptstyle \cong} & \downarrow{\scriptstyle \mathtt{upd}_B \times \mathtt{upd}_A ; \psi} \\ \mathbb{A}_A \times [\![A]\!] \times \mathbb{A}_B \otimes [\![B]\!] & \xrightarrow[\mathtt{upd}_A \times \mathtt{upd}_B ; \psi]{} & T1 \end{array}$$

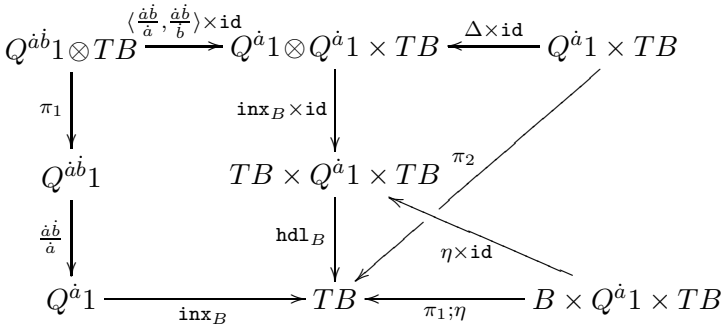
and also update and fresh-name are independent effects, that is,

$$\mathtt{new}_A \times \mathtt{upd}_B \,;\, \psi = \mathtt{new}_A \times \mathtt{upd}_B \,;\, \psi' \tag{SNR}$$

VII. $\mathcal{M}$ contains a natural transformation $\mathtt{inx} : K_{Q^{\dot{a}}1} \longrightarrow T$ for exception-inclusion, where $K_{Q^{\dot{a}}1}$ the constant $Q^{\dot{a}}1$ functor, such that each $\mathtt{inx}_B$ is an inner-component arrow and the following diagrams commute.

$$\begin{array}{ccc} A \times Q^{\dot{a}}1 & \xrightarrow{\mathtt{id} \times \mathtt{inx}_B} & A \times TB \\ {\scriptstyle \pi_2}\downarrow & & \downarrow{\scriptstyle \tau} \\ Q^{\dot{a}}1 & \xrightarrow[\mathtt{inx}_{A \times B}]{} & T(A \times B) \end{array} \qquad \begin{array}{ccc} Q^{\dot{a}}1 & \xrightarrow{\mathtt{inx}_{TB}} & T^2B \\ & {\scriptstyle \mathtt{inx}_B}\searrow & \downarrow{\scriptstyle \mu} \\ & & TB \end{array} \tag{NE1}$$

and for each object B an arrow $\mathtt{hdl}_B : TB \times Q^{\dot{a}}1 \times TB \longrightarrow TB$ for exception-handling such that the following commutes.



(NE2)

▲

The `new` transformation induces the following notion of name-abstraction for arrows. For any $f : Q^{\vec{a}a}A \longrightarrow TB$ we set

$$\langle\!\langle a \rangle\!\rangle f : Q^{\vec{a}}A \longrightarrow TB \triangleq Q^{\vec{a}}A \xrightarrow{\mathtt{new}} TQ^{\vec{a}a}A \xrightarrow{Tf} T^2B \xrightarrow{\mu} TB$$

We now give the semantics of $\nu\varepsilon\rho$ in a $\lambda_{\nu\varepsilon\rho}$-model.

Definition 8 Let $\langle \mathcal{M}, T, Q \rangle$ be a $\lambda_{\nu\varepsilon\rho}$-model. A typed term $\vec{a} \mid \Gamma \vdash M : A$ is mapped to an

arrow $[\![M]\!]_{\vec{a}\,|\,\Gamma} : Q^{\vec{a}}[\![\Gamma]\!] \longrightarrow T[\![A]\!]$, which we write simply as $[\![M]\!]$, in $\mathcal{M}$ as follows,

$$[\![\mathtt{skip}]\!] : Q^{\vec{a}}\Gamma \xrightarrow{Q^{\vec{a}}!;\, \frac{\vec{a}}{\epsilon}} 1 \xrightarrow{\eta} T1 \qquad [\![n]\!] : Q^{\vec{a}}\Gamma \xrightarrow{Q^{\vec{a}}!;\, \frac{\vec{a}}{\epsilon}} 1 \xrightarrow{n} \mathbb{N} \xrightarrow{\eta} T\mathbb{N}$$

$$[\![\ddot{a}]\!] : Q^{\vec{a}}\Gamma \xrightarrow{Q^{\vec{a}}!;\, \frac{\vec{a}}{\ddot{a}}} \mathbb{A}_A \xrightarrow{\eta} T\mathbb{A}_A \qquad [\![\mathtt{raise}\, \dot{a}]\!] : Q^{\vec{a}}\Gamma \xrightarrow{Q^{\vec{a}}!;\, \frac{\vec{a}}{\dot{a}}} Q^{\dot{a}}1 \xrightarrow{\mathtt{inx}_A} TA$$

$$\frac{[\![M]\!] : Q^{\vec{a}}(\Gamma \times A) \longrightarrow TB}{[\![\lambda x.M]\!] : Q^{\vec{a}}\Gamma \xrightarrow{\Lambda^T(\zeta';\, [\![M]\!])} A \mathrel{\tilde{\Rightarrow}} TB \xrightarrow{\eta} T(A \mathrel{\tilde{\Rightarrow}} TB)}$$

$$\frac{[\![M]\!] : Q^{\vec{a}}\Gamma \longrightarrow T(A \mathrel{\tilde{\Rightarrow}} TB) \qquad [\![N]\!] : Q^{\vec{a}}\Gamma \longrightarrow TA}{[\![M\, N]\!] : Q^{\vec{a}}\Gamma \xrightarrow{\langle [\![M]\!], [\![N]\!] \rangle;\, \psi} T((A \mathrel{\tilde{\Rightarrow}} TB) \times A) \xrightarrow{T\mathsf{ev}^T;\, \mu} TB}$$

$$\frac{[\![M]\!] : Q^{\vec{a}a}\Gamma \longrightarrow TA}{[\![\nu a.M]\!] = \langle a \rangle [\![M]\!] : Q^{\vec{a}}\Gamma \longrightarrow TA}$$

$$\frac{[\![M]\!] : Q^{\vec{a}}\Gamma \longrightarrow T\mathbb{A}_A \qquad [\![N]\!] : Q^{\vec{a}}\Gamma \longrightarrow T\mathbb{A}_A}{[\![[M = N]]\!] : Q^{\vec{a}}\Gamma \xrightarrow{\langle [\![M]\!], [\![N]\!] \rangle;\, \psi} T(\mathbb{A}_A \times \mathbb{A}_A) \xrightarrow{T\mathtt{eq}} T\mathbb{N}}$$

$$\frac{[\![M]\!] : Q^{\vec{a}}\Gamma \longrightarrow T\mathbb{A}_A \qquad [\![N]\!] : Q^{\vec{a}}\Gamma \longrightarrow TA}{[\![M := N]\!] : Q^{\vec{a}}\Gamma \xrightarrow{\langle [\![M]\!], [\![N]\!] \rangle;\, \psi} T(\mathbb{A}_A \times A) \xrightarrow{T\mathtt{upd}_A;\, \mu} T1}$$

$$\frac{[\![M]\!] : Q^{\vec{a}}\Gamma \longrightarrow T\mathbb{A}_A}{[\![!M]\!] : Q^{\vec{a}}\Gamma \xrightarrow{[\![M]\!]} T\mathbb{A}_A \xrightarrow{T\mathtt{drf}_A} T^2A \xrightarrow{\mu} TA}$$

$$\frac{[\![M]\!] : Q^{\vec{a}}\Gamma \longrightarrow TA \qquad [\![N]\!] : Q^{\vec{a}}\Gamma \longrightarrow TA}{\begin{array}{l}[\![\mathtt{try}\ M\ \mathtt{handle}\, \dot{a} \Rightarrow N]\!] : Q^{\vec{a}}\Gamma \xrightarrow{\langle [\![M]\!], Q^{\vec{a}}!;\, \frac{\vec{a}}{\dot{a}}, [\![N]\!] \rangle} TA \times Q^{\dot{a}}1 \times TA \\ \qquad\qquad \xrightarrow{\theta \times \mathtt{id};\, \tau'} T(TA \times Q^{a}1 \times TA) \xrightarrow{T\mathtt{hdl}_A;\, \mu} TA\end{array}}$$

plus standard translations for other term constructs. ▲

Observe the use of θ in the semantic translation of handling; the intuition is the following. θ separates the two components of the computation $[\![M]\!]$, and channels the outer component to the output and the inner component to the exception-handler.

The specifications of the $\lambda_{\nu\varepsilon\rho}$-model are tailored towards correctness. Let us write $S \models M \xrightarrow{\mathrm{r}} S' \models M'$ with r being a reduction rule different from CTX, if the last non-CTX rule in the related derivation is r. We write $M\,;N$ for $(\lambda d.N)M$, some d not in N, and relate to any store S the term $\bar{S}$ of type $\mathbb{1}$, by:

$$\bar{\epsilon} \triangleq \mathtt{skip}\,, \quad \overline{a, S} \triangleq \bar{S}\,, \quad \overline{\ddot{a} :: V, S} \triangleq (\ddot{a} := V\,;\bar{S})$$

Proposition 9 (Correctness) *For any typed term* $\vec{a} \mid \Gamma \vdash M : A$*, any* S *with* $\mathtt{dom}(S) = \vec{a}$ *and any* $\mathrm{r} \notin \{\mathrm{NEW}, \mathrm{UPD}, \mathrm{DRF}\}$*,*

- $S \vDash M \xrightarrow{\mathrm{r}} S \vDash M' \implies [\![M]\!] = [\![M']\!]$
- $S \vDash M \xrightarrow{\mathrm{UPD/DRF}} S' \vDash M' \implies [\![\bar{S}\,; M]\!] = [\![\bar{S}'\,; M']\!]$
- $S \vDash M \xrightarrow{\mathrm{NEW}} S, a \vDash M' \implies [\![\bar{S}\,; M]\!] = \langle a \rangle [\![\bar{S}\,; M']\!]$

Therefore, $S \vDash M \to S' \vDash M' \implies [\![\nu \vec{a}.(\bar{S}\,; M)]\!] = [\![\nu \vec{a}'.(\bar{S}'\,; M')]\!]$ *, with* $\mathtt{dom}(S') = \vec{a}'$ ■

Soundness doesn't follow from correctness; we also need computational adequacy. The latter is added as a specification.

Definition 10 (Adequacy) Let $\mathcal{M}$ be a $\lambda_{\nu\varepsilon\rho}$-model and $[\![\_]\!]$ the respective translation of $\nu\varepsilon\rho$. $\mathcal{M}$ is adequate if, for any typed term $\vec{a} \mid \varnothing \vdash M : \mathbb{N}$, if $[\![M]\!] = \langle \vec{b} \rangle [\![\bar{S}\,; 0]\!]$, some S, then there exists S' such that $\vec{a} \vDash M \longrightarrow\!\!\!\!\!\rightarrow S' \vDash 0$. ▲

Assuming now our running $\mathcal{M}$ is an adequate $\lambda_{\nu\varepsilon\rho}$-model we can easily show the following.

Proposition 11 (Equational Soundness)

$$[\![M]\!] = [\![N]\!] \implies M \lessapprox N$$

■

3.1.4 Completeness

To achieve completeness we need to introduce a preorder in the semantics to match the observational preorder of the syntax, as in (FA). This step, which is essentially a *quotienting* procedure, is found in many (but by no means all) fully abstract models based on game semantics.

Definition 12 (p-Observationality) An adequate $\lambda_{\nu\varepsilon\rho}$-model $\mathcal{M} = \langle \mathcal{M}, T, Q \rangle$ is p(reorder)-observational if, for all $\vec{a}$:

(I) There is an $O^{\vec{a}} \subseteq \mathcal{M}(Q^{\vec{a}}1, T\mathbb{N})$ such that for all $\vec{a} \mid \varnothing \vdash M : \mathbb{N}$,

$$[\![M]\!] \in O^{\vec{a}} \iff \exists S, \vec{b}.\, [\![M]\!] = \langle \vec{b} \rangle [\![\bar{S}\,; 0]\!]$$

(II) The induced intrinsic preorder on arrows in $\mathcal{M}(Q^{\vec{a}}A, TB)$ defined by $f \lesssim^{\vec{a}} g \iff$

$$\forall \rho : Q^{\vec{a}}(A \mathbin{\tilde{\Rightarrow}} TB) \to T\mathbb{N}.\, (\Lambda^{Q^{\vec{a}}}(f)\,; \rho \in O^{\vec{a}} \implies \Lambda^{Q^{\vec{a}}}(g)\,; \rho \in O^{\vec{a}})$$

with $\Lambda^{Q^{\vec{a}}}(f) : Q^{\vec{a}}1 \xrightarrow{\delta} Q^{\vec{a}}Q^{\vec{a}}1 \xrightarrow{Q^{\vec{a}}\Lambda^T(\zeta'\,; f)} Q^{\vec{a}}(A \mathbin{\tilde{\Rightarrow}} TB)$, satisfies, for all $a \# \vec{a}$ and relevant f, f',

$$f \lesssim^{\vec{a}a} f' \implies \langle a \rangle f \lesssim^{\vec{a}} \langle a \rangle f'$$
$$f \lesssim^{\vec{a}} f' \implies \frac{\vec{a}a}{\vec{a}}\,; f \lesssim^{\vec{a}a} \frac{\vec{a}a}{\vec{a}}\,; f'$$

We write $\mathcal{M}$ as $\langle \mathcal{M}, T, Q, O, \lesssim \rangle$. ▲

So, $O^{\vec{a}}$ contains those arrows that have a specific *observable behavior* in the model, and the semantic preorder is built over this notion. In particular, terms that yield 0 have observable behavior. The specifications of part (II) in the previous definition ensure that the intrinsic preorder is a congruence, i.e.

$$[\![M]\!] \lesssim [\![N]\!] \implies [\![\mathrm{C}[M]]\!] \lesssim [\![\mathrm{C}[N]]\!]$$

for all M, N and contexts C. Thus, assuming our running $\mathcal{M}$ is p-observational we can easily get one direction of (FA).

Lemma 13 (Inequational Soundness)

$$[\![M]\!] \lesssim [\![N]\!] \implies M \lessapprox N$$

■

In order to achieve completeness, and hence full-abstraction, we need to be able to express the intrinsic preorder solely by use of *definable test-arrows*.

Definition 14 (p-Definability) Let $\langle \mathcal{M}, T, Q, O, \lesssim \rangle$ be a p-observational $\lambda_{\nu\varepsilon\rho}$-model and let $[\![\_]\!]$ be the semantic translation of $\nu\varepsilon\rho$ to $\mathcal{M}$. $\mathcal{M}$ satisfies p-definability if, for any $\vec{a}, A, B$, there exists $D^{\vec{a}}_{A,B} \subseteq \mathcal{M}(Q^{\vec{a}}[\![A]\!], T[\![B]\!])$ such that:

- For each $f \in D^{\vec{a}}_{A,B}$ there exists term M such that $[\![M]\!] = f$,
- For each $f, g \in \mathcal{M}(Q^{\vec{a}}A, TB)$, $f \lesssim g$ iff

$$\forall \rho \in D^{\vec{a}}_{A \to B, \mathbb{N}} \,.\, (\Lambda^{Q^{\vec{a}}}(f)\,;\rho \in O^{\vec{a}} \implies \Lambda^{Q^{\vec{a}}}(g)\,;\rho \in O^{\vec{a}})$$

▲

For such a model $\mathcal{M}$ we achieve full abstraction; the proof of completeness is done by induction on the size of Γ, and the methodology is more or less standard.

Proposition 15 (FA) *For typed terms* $\vec{a} \mid \Gamma \vdash M, N : A$,

$$[\![M]\!] \lesssim [\![N]\!] \iff M \lessapprox N$$

■

4 The nominal games model

We build a model of $\nu\varepsilon\rho$ in a category of nominal arenas and strategies, following a route similar to that of [17]. The basic construction is $\mathcal{V}_{\mathtt{t}}$, the category of nominal arenas and total strategies. $\mathcal{V}_{\mathtt{t}}$ is constructed in $\mathbf{sNom}_{\mathrm{TYE}}$, and there are:

- for each type A an arena $\mathbb{A}_A$ for references to type A,

- an arena $\mathbb{A}_E$ for exceptions.

The translation $[\![A]\!]$ of a general type will make use of a *store arena* $\xi = \bigotimes_{A\in \mathrm{TY}}(\mathbb{A}_A \Rightarrow [\![A]\!])$, which will literally serve as a reference-store, and of the exception-arena $\mathbb{A}_E$. This will naturally lead us to a monadic semantics, with computation monad T defined on arenas by $TA = \xi \Rightarrow (A + \mathbb{A}_E) \otimes \xi$. Since arrow types involve the monad in their translation and the monad involves all types, we will have to first solve the domain equation:

$$\begin{aligned} [\![A \to B]\!] &= [\![A]\!] \stackrel{\sim}{\Rightarrow} (\xi \Rightarrow ([\![B]\!] + \mathbb{A}_E) \otimes \xi) \\ \xi &= \bigotimes\nolimits_A (\mathbb{A}_A \Rightarrow [\![A]\!]) \end{aligned} \tag{SE}$$

4.1 The category of nominal games

We assume a set of types TY and build our constructions inside $\mathbf{sNom}_{\mathrm{TYE}}$. We start with nominal arenas.

Definition 16 A nominal arena $A \triangleq (M_A, I_A, \vdash_A, \lambda_A)$ is given by:

- a strong nominal set M_A of moves,
- a nominal subset $I_A \subseteq M_A$ of initial moves,
- a nominal justification relation $\vdash_A \subseteq M_A \times \bar{I}_A$,
- a nominal labeling function $\lambda_A : M_A \to \{O, P\} \times \{A, Q\}$.

Moves in M_A are denoted by m_A and variants, and initial moves by i_A and variants. By $\bar{I}_A$ we denote $M_A \setminus I_A$. λ_A labels moves as Opponent or Player moves, and as Answers or Questions.
An arena A satisfies also the conditions:

(f) For each $m \in M_A$, there exists unique $k \geq 0$ such that $I_A \ni m_1 \vdash_A \cdots \vdash_A m_k \vdash_A m$, for some m_l's in M_A.
k is called the level of m, so initial moves have level 0.

(l1) Initial moves are P-answers.

(l2) If $m_1, m_2 \in M_A$ are at consecutive levels then λ_A assigns them complementary OP-labels.

(l3) Answers may only justify Questions.

A *prearena* is an arena with its initial moves labeled OQ. Given arenas A and B, construct the prearena $A \to B$ as:

$$\begin{aligned} M_{A\to B} &\triangleq M_A + M_B \\ I_{A\to B} &\triangleq I_A \\ \lambda_{A\to B} &\triangleq [(i_A \mapsto OQ\ ,\ m_A \mapsto \overline{\lambda_A}(m_A))\ ,\ \lambda_B] \\ \vdash_{A\to B} &\triangleq \{(i_A, i_B)\} \cup \{(m, n) \mid m \vdash_{A,B} n\} \end{aligned}$$

▲

Because of condition (f), arenas can be represented by directed connected graphs with no directed cycles. From arenas A, B we can construct the following arenas.

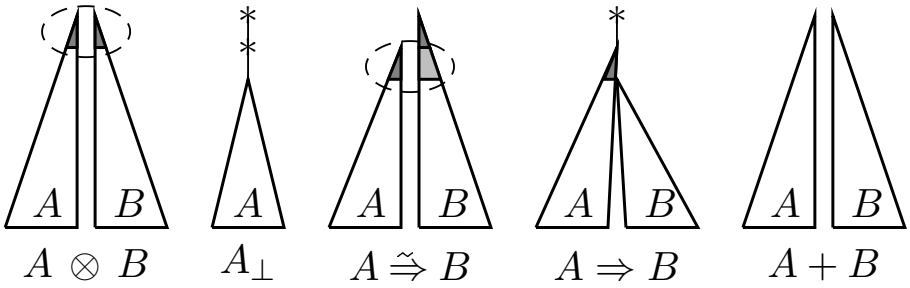


Figure 1: Basic arena constructions

The constructions are defined formally in [17]; for example,

$$\begin{array}{rcll} M_{A\otimes B} & \triangleq & I_A{\times}I_B + \bar{I}_A + \bar{I}_B & (A \otimes B) \\ I_{A\otimes B} & \triangleq & I_A{\times}I_B & \\ \lambda_{A\otimes B} & \triangleq & [((i_A, i_B) \mapsto PA)\,, \lambda_A\,, \lambda_B] & \\ \vdash_{A\otimes B} & \triangleq & \{((i_A, i_B), m) \mid i_A \vdash_A m \vee i_B \vdash_B m\} \cup (\vdash_A\restriction \bar{I}_A{}^2) \cup (\vdash_B\restriction \bar{I}_B{}^2) & \end{array}$$

$$\begin{array}{rcll} M_{A+B} & \triangleq & M_A + M_B & (A + B) \\ I_{A+B} & \triangleq & I_A + I_B & \\ \lambda_{A+B} & \triangleq & [\lambda_A\,, \lambda_B] & \\ \vdash_{A+B} & \triangleq & \vdash_A \cup \vdash_B & \end{array}$$

The simplest arena is $0 \triangleq (\varnothing, \varnothing, \varnothing, \varnothing)$. Other (flat) arenas are 1, $\mathbb{N}$ and $\mathbb{A}^{\vec{a}}$, for any $\vec{a} \in \mathbb{A}^{\#}$, defined as follows.

$$M_{\mathbb{N}} = I_{\mathbb{N}} \triangleq \mathbb{N} \qquad M_1 = I_1 \triangleq \{*\} \qquad M_{\mathbb{A}^{\vec{a}}} = I_{\mathbb{A}^{\vec{a}}} \triangleq \mathbb{A}^{\vec{a}}$$

In case $\vec{a}$ is singleton, the last construction above yields arenas $\mathbb{A}_A$, each type A, and $\mathbb{A}_E$. We will usually identify graph-isomorphic arenas related by isomorphisms which simply manipulate $*$'s; for example, for any A, B,

$$0 + A = A + 0 = A \quad , \quad 1 \mathrel{\tilde{\Rightarrow}} A = A \quad , \quad A \Rightarrow B = A \mathrel{\tilde{\Rightarrow}} B_{\perp}$$

Of the previous constructors all look familiar apart from $\mathrel{\tilde{\Rightarrow}}$. The latter can be seen as a function-space constructor merging the contravariant part of its RHS with its LHS. For example, for any A, B, C, we have

$$A \mathrel{\tilde{\Rightarrow}} \mathbb{N} = \mathbb{N} \quad \text{and} \quad A \mathrel{\tilde{\Rightarrow}} (B \Rightarrow C) = (A{\otimes}B) \Rightarrow C$$

In the first case we see that the $\mathbb{N}$ which appears on the RHS of $\tilde{\Rightarrow}$ has no contravariant part, and hence A is redundant. In the second case, the contravariant part of $B \Rightarrow C$, i.e. B, is merged with A.

We move on to describe nominal games. Nominal games are played in prearenas using moves which are attached with name-lists capturing name-environments.

Definition 17 A ***move-with-names*** of a (pre)arena A is a pair, written $m^{\vec{a}}$, where m is a move of A and $\vec{a}$ is a finite list of distinct names, a name-list. We set $\mathsf{nlist}(m^{\vec{a}}) \triangleq \vec{a}$. ▲

Note that moves-with-names have strong support. We need to introduce some further notation for sequences.

Notation 18 (Sequences) A sequence s will be usually denoted by $xy\ldots$, where $x, y, ...$ are the elements of s. For sequences s, t,

- if $s \leq t$ then $t = s(t - s)$,
- s^- denotes s with its last element removed,
- if $s = s_1 \cdots s_n$ then s_1 is the first element of s and s_n the last. Moreover,
  - n is the *length* of s, and is denoted by $|s|$,
  - $s.i$ denotes s_i and $s.\text{-}i$ denotes s_{n+1-i}, that is the i-th element from the tail of s (for example, $s.\text{-}1$ in s_n),
  - $s_{\leq s_i}$ denotes $s_1 \cdots s_i$, and so does $s_{<s_{i+1}}$.
- if s is a sequence of moves-with-names then we denote by $\underline{s}$ its *underlying sequence*, that is the sequence retrieved from s by deleting all of its name-lists, in which case $s = \underline{s}^{\mathsf{nlist}(s)}$. ▲

We proceed to defining plays. A ***justified sequence*** over a prearena A is a finite sequence s of OP-alternating moves such that, except for $s.1$ which is initial, every move $s.i$ has a *justification pointer* to some $s.j$ such that $j < i$ and $s.j \vdash_A s.i$; we say that $s.j$ *(explicitly) justifies* $s.i$. The ***P-view***, $\ulcorner s \urcorner$, of a justified sequence s is:

$$\begin{array}{rl} \ulcorner sx \urcorner \triangleq \ulcorner s \urcorner x & \text{if } x \text{ a P-move} \\ \ulcorner x \urcorner \triangleq x & \text{if } x \text{ is initial} \\ \ulcorner sxs'y \urcorner \triangleq \ulcorner s \urcorner xy & \text{if } y \text{ an O-move justified by } x \end{array}$$

Definition 19 Let A be a prearena. A ***legal sequence*** on A is a justified sequence of moves-with-names that satisfies Visibility and Well-Bracketing (v. [12, 8]). A legal sequence s is a ***play*** if $s.1$ has empty name-list and s also satisfies the following Name Change Conditions:

(NC1) The name-list of a P-move x in s contains as a prefix the name-list of its preceding O-move. It possibly contains some other names, all of which are fresh for $s_{<x}$.

(NC2′) Any name in the support of a P-move x in s that is fresh for $\ulcorner s_{<x} \urcorner$ is contained in the name-list of x.

(NC3) The name-list of a non-initial O-move in s is that of the P-move explicitly justifying it.

The set of plays on a prearena A is denoted by P_A. ▲

Thus, here we take plays to be *innocent ϵ-plays* in terms of [17]. A name a ***is introduced*** (by Player) in a play s, written $a \in \mathcal{L}(s)$, if there exist consecutive moves yx in s such that x is a P-move and $a \in \mathtt{S}(\mathsf{nlist}(x) - \mathsf{nlist}(y))$.

From plays we move on to (innocent) strategies.

Definition 20 A ***strategy*** σ is a set of equivalence classes $[s]$ of plays satisfying *prefix closure*, *contingency completeness*, *determinacy* and *innocence*:

- If $[su] \in \sigma$ then $[s] \in \sigma$.
- If even-length $[s] \in \sigma$ and sx is a play then $[sx] \in \sigma$.
- If even-length $[s_1x_1], [s_2x_2] \in \sigma$ and $[s_1] = [s_2]$ then $[s_1x_1] = [s_2x_2]$.
- If $[s_1x_1], [s_2] \in \sigma$, s_1 odd-length and $[\ulcorner s_1 \urcorner] = [\ulcorner s_2 \urcorner]$ then there exists some $[s_2x_2] \in \sigma$ such that $[\ulcorner s_1x_1 \urcorner] = [\ulcorner s_2x_2 \urcorner]$. ▲

Some basic strategies are the following.

Definition 21 For any $\vec{b} \preceq \vec{a} \in \mathbb{A}^{\#}$, any $n \in \mathbb{N}$ and any arena B, define the following strategies.

- $(\frac{\vec{a}}{\vec{b}})_1 : \mathbb{A}^{\vec{a}} \to \mathbb{A}^{\vec{b}} \triangleq \{[\vec{a}\,\vec{b}]\}$
- $\mathtt{eq}_A : \mathbb{A}_A \otimes \mathbb{A}_A \to \mathbb{N} \triangleq \{[(\ddot{a}, \ddot{a})\,0\,], [(\ddot{a}, \ddot{b})\,1\,] \mid \ddot{a} \# \ddot{b}\}$
- $n : 1 \to \mathbb{N} \triangleq \{[*\,n\,]\}$
- $!_B : B \to 1 \triangleq \{[i_B\,*]\}$
- $\mathtt{id}_B : B \to B \triangleq \{[s\,x\,x] \mid [s] \in \mathtt{id}_B \wedge s\,x \in P_{B \to B}\}$ ▲

Plays are composed as usually in game semantics, that is by parallel composition and hiding, with some extra care taken for fresh names.

Definition 22 (Composable plays) For any $s \in P_{A \to B}$, $t \in P_{B \to C}$, s and t are *almost composable*, $s \smile t$, if $\underline{s} \upharpoonright B = \underline{t} \upharpoonright B$.
s and t are *composable*, $s \asymp t$, if $s \smile t$ and, for any $s' \leq s$, $t' \leq t$ with $s' \smile t'$,

(C1) If s' ends in a P-move in A introducing some name a then $a \# t'$; dually, if t' ends in a P-move in C introducing some name a then $a \# s'$.

(C2) If both s', t' end in B and s' ends in a P-move introducing some name a then $a \# t'^{-}$; dually, if t' ends in a P-move introducing some name a then $a \# s'^{-}$. ▲

If $s \in P_{A\to B}$ and $t \in P_{B\to C}$ with $s \smile t$ then either $s \upharpoonright B = t = \epsilon$, or s ends in A and t in B, or s ends in B and t in C, or both s and t end in B (*Zipper Lemma*). Hence, composable plays are composed as below.

Definition 23 ($\| \bullet$;) Let $s \in P_{A\to B}$ and $t \in P_{B\to C}$ with $s \asymp t$. Their parallel interaction $s \parallel t$ and their mix $s \bullet t$, which returns the final name-list in $s \parallel t$, are defined as below.

$$sm_A^{\vec{b}} \parallel t \triangleq (s \parallel t) m_A^{sm_A^{\vec{b}} \bullet t} \qquad sm_B^{\vec{b}} \parallel tm_B^{\vec{c}} \triangleq (s \parallel t) m_B^{sm_B^{\vec{b}} \bullet tm_B^{\vec{c}}}$$

$$s \parallel tm_C^{\vec{c}} \triangleq (s \parallel t) m_C^{s \bullet tm_C^{\vec{c}}} \qquad \epsilon \parallel \epsilon \triangleq \epsilon \qquad \epsilon \bullet \epsilon \triangleq \epsilon$$

$$sm_{A(O)}^{\vec{b}} \bullet t \triangleq \vec{b}' \qquad sm_{B(P)}^{\vec{b}} \bullet tm_{B(O)}^{\vec{c}} \triangleq (s \bullet t), \vec{b}''$$

$$sm_{A(P)}^{\vec{b}} \bullet t \triangleq (s \bullet t), \vec{b}'' \qquad sm_{B(O)}^{\vec{b}} \bullet tm_{B(P)}^{\vec{c}} \triangleq (s \bullet t), \vec{c}''$$

$$s \bullet tm_{C(P)}^{\vec{c}} \triangleq (s \bullet t), \vec{c}'' \qquad s \bullet tm_{C(O)}^{\vec{c}} \triangleq \vec{c}'$$

where $\vec{b}''$ is $\vec{b} - \mathsf{nlist}(s\text{-}1)$ and $\vec{b}'$ is the name-list of $m_{A(O)}$'s justifier in $s \parallel t$, and similarly for $\vec{c}', \vec{c}''$.
The composite of s and t is: $s\,;t \triangleq (s \parallel t) \upharpoonright AC$.

For strategies $\sigma : A \to B$ and $\tau : B \to C$, their composition is: $\sigma\,;\tau \triangleq \{\, [s\,;t] \mid [s] \in \sigma \wedge [t] \in \tau \wedge s \asymp t \,\}$. ▲

We can prove the following.

Proposition 24 *If $s \in P_{A\to B}$ and $t \in P_{B\to C}$ with $s \asymp t$, then $s\,;t \in P_{A\to C}$.*
If $\sigma : A \to B$ and $\tau : B \to C$ are strategies then so is $\sigma\,;\tau$. Moreover, strategy-composition is associative and composition with `id` *is identity.* ■

Hence, strategies compose and form a category.

Definition 25 $\mathcal{V}$ is the category having nominal arenas as objects and (innocent nominal) strategies as arrows. ▲

An easy construction in $\mathcal{V}$ is that of coproducts.

Proposition 26 *For any objects A, B in $\mathcal{V}$, the triple $(\mathtt{in}_1, \mathtt{in}_2, A + B)$, where*

$$\mathtt{in}_1 : A \longrightarrow A + B \triangleq \{[i_A\, i_A\, s] \mid [i_A\, i_A\, s] \in \mathtt{id}_A\}$$
$$\mathtt{in}_2 : B \longrightarrow A + B \triangleq \{[i_B\, i_B\, s] \mid [i_B\, i_B\, s] \in \mathtt{id}_B\}$$

is a coproduct. For any $A \xrightarrow{f} C \xleftarrow{g} B$ we have

$$[f, g] : A + B \longrightarrow C \triangleq \{[i_A\, i_C\, s] \mid [i_A\, i_C\, s] \in f\} \cup \{[i_B\, i_C\, s] \mid [i_B\, i_C\, s] \in g\}$$

■

We find useful to represent strategies by their viewfunctions.

Definition 27 A ***viewfunction*** f is a set of equivalence classes of plays that are even-length P-views, which satisfies *even-prefix closure* and *single-valuedness*:

- If $[s] \in f$ and t is an even-length prefix of s then $[t] \in f$.
- If $[s_1 x_1], [s_2 x_2] \in f$ and $[s_1] = [s_2]$ then $[s_1 x_1] = [s_2 x_2]$. ▲

There are maps `viewf` and `strat` from strategies to viewfunctions and viceversa such that

$$f = \texttt{viewf}(\texttt{strat}(f)) \quad \wedge \quad \sigma = \texttt{strat}(\texttt{viewf}(\sigma))$$

From now on, will be defining strategies via their viewfunctions.

4.2 Semantics in $\mathcal{V}_{\mathtt{t}}$

We define several subclasses of innocent strategies, with regard on initial and level-1 moves. For an arena A we write J_A for its set of level-1 moves, and we denote the latter by j_A and variants.

Definition 28 An innocent strategy $\sigma : A \longrightarrow B$ is ***total*** if for any $[i_A] \in \sigma$ there exists $[i_A\, i_B] \in \sigma$.
A total strategy $\sigma : A \longrightarrow B$ is ***ttotal*** if for any $[i_A\, i_B\, j_B] \in \sigma$ there exists $[i_A\, i_B\, j_B\, j_A] \in \sigma$, and whenever $[s\, x\, j'^{\vec{a}'}_A] \in f$ then $x \in J_B$.
$\mathcal{V}_{\mathtt{t}}$ is the lluf subcategory of $\mathcal{V}$ whose arrows are total strategies, and $\mathcal{V}_{\mathtt{tt}}$ the lluf subcategory of $\mathcal{V}_{\mathtt{t}}$ of ttotal strategies. ▲

Henceforth, by strategies we shall mean total strategies, unless stated otherwise. Now, the constructions of figure 1 have arrow-counterparts. Let $f : A \to A'$, $g : B \to B'$ in $\mathcal{V}_{\mathtt{t}}$ and $h : B \to B'$ in $\mathcal{V}_{\mathtt{tt}}$, then

in $f_\perp : A_\perp \to A'_\perp$ Player initially plays a sequence of asterisks $[*_1\ *'_1\ *'_2\, *_2\,]$ and then continues playing like f.

in $f \otimes g : A \otimes B \to A' \otimes B'$ Player answers initial moves $[(i_A, i_B)\,]$ with f's answer to $[i_A\,]$ and g's answer to $[i_B\,]$. Then, according to whether Opponent plays in $J_{A'}$ or in $J_{B'}$, Player plays like f or like g respectively.

in $f \mathbin{\hat{\Rightarrow}} h : A' \mathbin{\hat{\Rightarrow}} B \to A \mathbin{\hat{\Rightarrow}} B'$ Player answers initial moves $[i_B\,]$ like h and then responds to $[i_B\, i_{B'}\, (i_A, j_{B'})\,]$ with f's answer to $[i_A\,]$ and h's response to $[i_B\, i_{B'}\, j_{B'}\,]$ (hence the need for ttotality of h). It then plays like f or like h, according to Opponent's next move.

We can also define infinite tensor products of pointed arenas, where an arena A is ***pointed*** if I_A is singleton (in which case the unique initial move is necessarily equivariant). For pointed arenas $\{A_i\}_{i\in\omega}$ construct their product $\bigotimes_i A_i$ by 'gluing them together'

at their initial moves. Since these are equivariant, the resulting initial move is also equivariant, and we denote it by "$*$". For any pointed A_i's and B_i's and any $\{f_i : A_i \to B_i\}_{i\in\omega}$ define:

$$\bigotimes_i f_i \triangleq \mathtt{strat}\{[* * s] \mid \exists k.\, [i_{A_k}\, i_{B_k}\, s] \in \mathtt{viewf}(f_k)\}$$

Take $\mathcal{V}_{\mathtt{t}*}$ to be the full subcategory of $\mathcal{V}_{\mathtt{t}}$ of pointed arenas.

Our constructions enjoy the following properties.

Proposition 29 *All of the following are functors.*

$$\_\otimes\_ : \mathcal{V}_{\mathtt{t}} \times \mathcal{V}_{\mathtt{t}} \to \mathcal{V}_{\mathtt{t}}\ ,\ \ \_\tilde{\Rightarrow}\_ : (\mathcal{V}_{\mathtt{t}})^{\mathrm{op}} \times \mathcal{V}_{\mathtt{tt}} \to \mathcal{V}_{\mathtt{tt}}$$
$$(\_)_\perp : \mathcal{V}_{\mathtt{t}} \to \mathcal{V}_{\mathtt{tt}} \quad , \quad \bigotimes\_ : \textstyle\prod_{i\in\omega}\mathcal{V}_{\mathtt{t}*} \to \mathcal{V}_{\mathtt{t}*}$$

Moreover, $\mathcal{V}_{\mathtt{t}}$ is a symmetric monoidal category under $\otimes$, and is partially closed in the following sense. For any object B, the functor $\_\otimes B : \mathcal{V}_{\mathtt{t}} \to \mathcal{V}_{\mathtt{t}}$ has a partial right adjoint $B \tilde{\Rightarrow} \_ : \mathcal{V}_{\mathtt{t}*} \to \mathcal{V}_{\mathtt{t}}$, that is for any object A and any pointed object C there exists a bijection

$$\Lambda^B_{A,C} : \mathcal{V}_{\mathtt{t}}(A\otimes B, C) \xrightarrow{\cong} \mathcal{V}_{\mathtt{t}}(A, B \tilde{\Rightarrow} C)$$

natural in A, C. Moreover, 1 is a terminal object and $\otimes$ is a product constructor in $\mathcal{V}_{\mathtt{t}}$, so $\mathcal{V}_{\mathtt{t}}$ has finite products. Finally, $\mathcal{V}_{\mathtt{t}}$ inherits coproducts from $\mathcal{V}$ and is distributive. ■

Now, the full form of the store equation (SE) is the following.

$$[\![\mathbb{1}]\!] = 1 \qquad [\![\mathbb{N}]\!] = \mathbb{N} \qquad [\![[A]]\!] = \mathbb{A}_A \qquad [\![A\otimes B]\!] = [\![A]\!]\otimes[\![B]\!]$$

$$[\![A \to B]\!] = [\![A]\!] \tilde{\Rightarrow} (\xi \Rightarrow ([\![B]\!] + \mathbb{A}_E) \otimes \xi) \quad \xi = \bigotimes_A(\mathbb{A}_A \Rightarrow [\![A]\!])$$

We solve it by upgrading it to a recursive functor equation and then recurring to minimal-invariants theory for games (v. [12]). The method is more or less standard and is applied in some length in [17]. Restricting our attention to objects, the solution we get is the least fixpoint of the function on arenas induced by (SE), if we order arenas by the subset ordering.

Definition 30 (ξ, $\circledast$ and $[\![A]\!]$) ξ and $[\![A]\!]$, for each type A, are defined via the least fixpoint solution of (SE).
ξ is pointed; we denote its unique initial move by $\circledast$. ▲

We proceed in constructing a $\lambda_{\nu\varepsilon\rho}$-model $\mathcal{V}_{\mathtt{t}} = \langle \mathcal{V}_{\mathtt{t}}, T, Q\rangle$. First we define the family of local-state comonads $Q^{\vec{a}}$.

Definition 31 (Local-state comonads) For each $\vec{a} \in \mathbb{A}^{\#}$ take $(Q^{\vec{a}}, \varepsilon, \delta)$ to be the product comonad with basis $\mathbb{A}^{\vec{a}}$, that is

- $Q^{\vec{a}} : \mathcal{V}_{\mathtt{t}} \to \mathcal{V}_{\mathtt{t}} \triangleq \mathbb{A}^{\vec{a}}\otimes\_$
- $\varepsilon : Q^{\vec{a}} \to Id_{\mathcal{V}_{\mathtt{t}}} \triangleq \{\varepsilon_A : \mathbb{A}^{\vec{a}}\otimes A \xrightarrow{\pi_2} A\}$

- $\delta : Q^{\vec{a}} \longrightarrow (Q^{\vec{a}})^2 \triangleq \{\delta_A : \mathbb{A}^{\vec{a}} \otimes A \xrightarrow{\Delta \otimes \mathtt{id}} \mathbb{A}^{\vec{a}} \otimes \mathbb{A}^{\vec{a}} \otimes A\}$

For each $\vec{a} \preceq \vec{a}$ define the natural transformation $\frac{\vec{a}}{\vec{a}'} : Q^{\vec{a}} \longrightarrow Q^{\vec{a}'}$ by taking,

- $(\frac{\vec{a}}{\vec{a}'})_A : \mathbb{A}^{\vec{a}} \otimes A \longrightarrow \mathbb{A}^{\vec{a}'} \otimes A \triangleq (\frac{\vec{a}}{\vec{a}'})_1 \otimes \mathtt{id}_A$
- $(\frac{\vec{a}}{\vec{a}'})_1 : \mathbb{A}^{\vec{a}} \longrightarrow \mathbb{A}^{\vec{a}'} \triangleq \{[\vec{a}\,\vec{a}']\}$ ▲

Proposition 32 *For each $\vec{a} \in \mathbb{A}^{\#}$, the triple $(Q^{\vec{a}}, \varepsilon, \delta)$ forms a comonad over $\mathcal{V}_{\mathtt{t}}$. Moreover, items* (Va,b,c,d) *of definition 7 are satisfied.* ■

We proceed to constructing the monad T, by composing a store monad $\ddot{T}$ with an exception monad $\dot{T}$. For the composition to be a strong monad, a distributivity law (v. [3]) is needed.

Definition 33 Let $(\ddot{T}, \ddot{\eta}, \ddot{\mu}, \ddot{\tau})$ and $(\dot{T}, \dot{\eta}, \dot{\mu}, \dot{\tau})$ be monads over a category $\mathcal{C}$. A distributive law of $\dot{T}$ over $\ddot{T}$ is a natural transformation $\ell : \dot{T}\ddot{T} \longrightarrow \ddot{T}\dot{T}$ such that, for any A, B,

- $\dot{T}\ddot{\eta}_A \,;\ell_A = \ddot{\eta}_{\dot{T}A}$ and $\dot{\eta}_{\ddot{T}A} \,;\ell_A = \ddot{T}\dot{\eta}_A$,
- $\dot{T}\ddot{\mu}_A \,;\ell_A = \ell_{\ddot{T}A} \,;\ddot{T}\ell_A \,;\ddot{\mu}_{\dot{T}A}$ and $\dot{\mu}_{\ddot{T}A} \,;\ell_A = \dot{T}\ell_A \,;\ell_{\dot{T}A} \,;\ddot{T}\dot{\mu}_A$
- $\dot{\tau}_{A,\ddot{T}B} \,;\dot{T}\ddot{\tau}_{A,B} \,;\ell_{A\otimes B} = \mathtt{id}\otimes\ell_B \,;\ddot{\tau}_{A,\dot{T}B} \,;\ddot{T}\tau_{A,B}$

If such a distributive law exists then we can define the ***compound monad*** (T, η, μ, τ) as,

- $T \triangleq \ddot{T}\dot{T}$
- $\eta_A : A \xrightarrow{\ddot{\eta}_A} \ddot{T}A \xrightarrow{\ddot{T}\dot{\eta}_A} TA$
- $\mu_A : T^2A \xrightarrow{\ddot{T}\ell_{\dot{T}A}} \ddot{T}^2\dot{T}^2A \xrightarrow{\ddot{\mu}_{\dot{T}^2A}} \ddot{T}\dot{T}^2A \xrightarrow{\ddot{T}\dot{\mu}_A} TA$
- $\tau_{A,B} : A\otimes TB \xrightarrow{\ddot{\tau}_{A,\dot{T}_B}} \ddot{T}(A\otimes\dot{T}B) \xrightarrow{\ddot{T}\dot{\tau}_{A,B}} T(A\otimes B)$ ▲

In [3] it is shown that if there exists a distributive law of $\dot{T}$ over $\ddot{T}$ satisfying the first two equations then $\ddot{T}\dot{T}$ is a monad. It is straightforward to see that the equation diagram makes $\ddot{T}\dot{T}$ a strong monad. We can also show the following.

Lemma 34 *Let T be a compound monad as above. T is precompound with θ defined by*

$$\theta_A : TA \longrightarrow T^2A \triangleq \ddot{T}\dot{\eta}_{\dot{T}A} \,;T\ddot{\eta}_{\dot{T}A}$$

■

The inner component $\dot{T}$ of our computational monad T is an exception monad, defined by use of coproducts.

Definition 35 Define the quadruple $(\dot{T}, \dot{\eta}, \dot{\mu}, \dot{\tau})$ as follows.

- $\dot{T} : \mathcal{V}_{\mathtt{t}} \to \mathcal{V}_{\mathtt{t}} \triangleq \_ + \mathbb{A}_E$

- $\dot{\eta}_A : A \longrightarrow \dot{T}A \triangleq \mathtt{in}_1$
- $\dot{\mu}_A : \dot{T}^2 A \longrightarrow \dot{T}A \triangleq [\mathtt{id}, \mathtt{in}_2]$
- $\dot{\tau}_{A,B} : A \otimes \dot{T}B \longrightarrow \dot{T}(A \otimes B) \triangleq \mathtt{dst}_{A,B,E} ; (\mathtt{id} + \pi_2)$ ▲

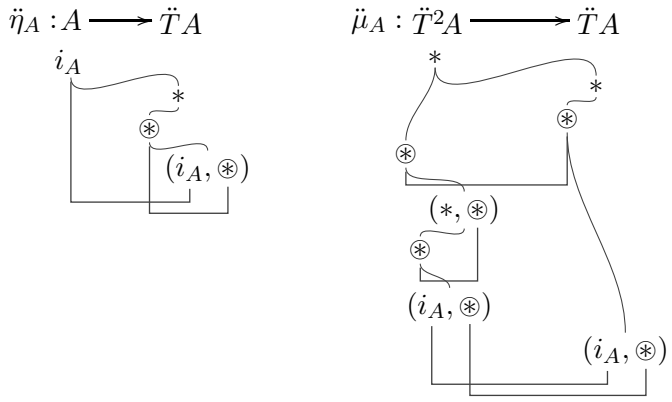
The outer component $\ddot{T}$ is a store monad, defined by use of currying and of the lifting monad. The lifting monad is a strong monad $((\_)_\perp, \mathtt{up}, \mathtt{dn}, \mathtt{st})$ given by a standard construction. It yields a λ_c-model by taking, for each A, B, C,

$$(C_\perp)^B \triangleq B \mathrel{\tilde{\Rightarrow}} C_\perp \quad \text{and} \quad \Lambda^\perp(A \otimes B, C_\perp) \triangleq \Lambda(A \otimes B, C_\perp)$$

Definition 36 Define the quadruple $(\ddot{T}, \ddot{\eta}, \ddot{\mu}, \ddot{\tau})$ as follows.

- $\ddot{T} : \mathcal{V}_\mathtt{t} \to \mathcal{V}_\mathtt{t} \triangleq \xi \Rightarrow (\_ \otimes \xi) = \xi \mathrel{\tilde{\Rightarrow}} (\_ \otimes \xi)_\perp$
- $\ddot{\eta}_A : A \longrightarrow \dot{T}A \triangleq \Lambda(\tilde{\eta}_A)$
- $\tilde{\eta}_A : A \otimes \xi \xrightarrow{\mathtt{up}} (A \otimes \xi)_\perp$
- $\ddot{\mu}_A : \dot{T}^2 A \longrightarrow \dot{T}A \triangleq \Lambda(\tilde{\mu}_A)$
- $\tilde{\mu}_A : \ddot{T}^2 A \otimes \xi \xrightarrow{\mathtt{ev}} (\ddot{T}A \otimes \xi)_\perp \xrightarrow{\mathtt{ev}_\perp} (A \otimes \xi)_{\perp\perp} \xrightarrow{\mathtt{dn}} (A \otimes \xi)_\perp$
- $\dot{\tau}_{A,B} : A \otimes \dot{T}B \longrightarrow \dot{T}(A \otimes B) \triangleq \Lambda(\tilde{\tau}_{A,B})$
- $\tilde{\tau}_{A,B} : (A \otimes \ddot{T}B) \otimes \xi \xrightarrow{\cong ; \mathtt{id} \otimes \mathtt{ev}} A \otimes (B \otimes \xi)_\perp \xrightarrow{\mathtt{st} ; \cong} ((A \otimes B) \otimes \xi)_\perp$. ▲

For example, $\ddot{\eta}$ and $\ddot{\mu}$ are given concretely as below.[4]



It is not difficult to show the following.

[4] In the diagrams we use curved lines for justification pointers; polygonic lines denote that the strategy copycats between the connected moves.

Proposition 37 *Both* $(\dot{T}, \dot{\eta}, \dot{\mu}, \dot{\tau})$ *and* $(\ddot{T}, \ddot{\eta}, \ddot{\mu}, \ddot{\tau})$ *are strong monads over* $\mathcal{V}_{\mathtt{t}}$*. Moreover,* $\dot{T}$ *distributes over* $\ddot{T}$*, with distributivity transformation* $\ell_A : \dot{T}\ddot{T}A \longrightarrow \ddot{T}\dot{T}A \triangleq \Lambda(\tilde{\ell}_A)$*, where,*

$$\begin{aligned}\tilde{\ell}_A &: ((\xi \Rightarrow A \otimes \xi) + \mathbb{A}_E) \otimes \xi \longrightarrow ((A + \mathbb{A}_E) \otimes \xi)_\perp \\ &\triangleq \mathtt{dst}\,;\mathtt{ev}{+}\mathtt{id}\,;[\mathtt{in}_{1\perp}, \mathtt{in}_2; \mathtt{up}]\,;(\mathtt{dst}^{-1})_\perp\end{aligned}$$

and $\mathtt{dst}$ *the distributivity transformation of* $\otimes$ *over* $+$*.*
Hence, by composing $\dot{T}$ *with* $\ddot{T}$ *we obtain a strong monad* (T, η, μ, τ)*, as in definition 33.* T *yields a* λ_c*-model over* $\mathcal{V}_{\mathtt{t}}$ *by taking,*

$$(TC)^B \triangleq B \mathrel{\dot{\Rightarrow}} TC \quad \textit{and} \quad \Lambda^T(A \otimes B, TC) \triangleq \Lambda(A \otimes B, TC)$$

since TC *is always pointed.* ■

We proceed in defining the $\mathtt{new}$ transformation.

Definition 38 For each $\vec{a}a \in \mathbb{A}^\#$ define the transformation $\mathtt{new}^{\vec{a}a} : Q^{\vec{a}} \longrightarrow TQ^{\vec{a}a}$ as follows,

- $\mathtt{new}^{\vec{a}a}_A : Q^{\vec{a}}A \longrightarrow TQ^{\vec{a}a}A \triangleq \mathbb{A}^{\vec{a}} \otimes A \xrightarrow{\mathtt{new}^{\vec{a}a}_1 \otimes \mathtt{id}\,;\,\tau'} T(\mathbb{A}^{\vec{a}a} \otimes A)$
- $\mathtt{new}^{\vec{a}a}_1 : Q^{\vec{a}}1 \longrightarrow TQ^{\vec{a}a}1 \triangleq \mathtt{strat}\{[(\vec{a}, *) \ast \circledast (\vec{a}a, \circledast)^a\, s] \mid [\circledast \circledast s] \in \mathtt{viewf}(\mathtt{id}_\xi)\}$. ▲

Following the convention described right after definition 7 we can define an arrow $\langle a \rangle f : Q^{\vec{a}}A \longrightarrow TB$ for each $f : Q^{\vec{a}a}A \longrightarrow TB$. Concretely, the construction is given by taking

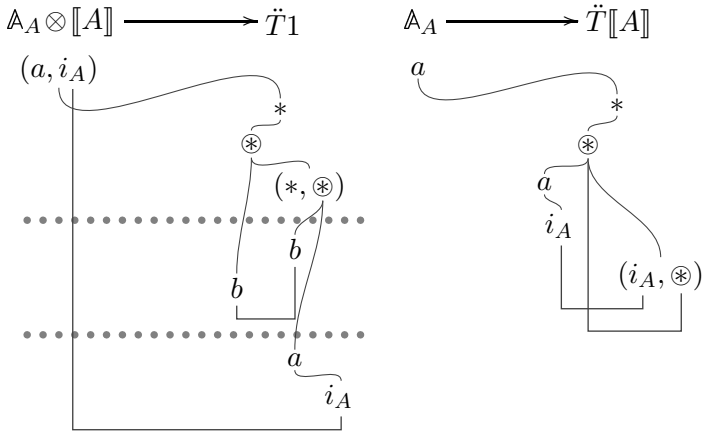
$$\langle a \rangle f \triangleq \mathtt{strat}\{[(\vec{a}, i_A) \ast \circledast m^{a\vec{b}} s] \mid [(\vec{a}a, i_A) \ast \circledast m^{\vec{b}} s] \in \mathtt{viewf}(f) \wedge a \# i_A\}$$

More than that, the above construction generalises to $f : Q^{\vec{a}a}A \longrightarrow B'$ for *any pointed* B', in a straightforward manner. We can then show the following.

Proposition 39 *If* C *pointed and* $f : Q^{\vec{a}a}(A \otimes B) \longrightarrow C$ *then* $\Lambda(\bar{\zeta}'\,; \langle a \rangle f) = \langle a \rangle \Lambda(\bar{\zeta}'\,; f)$*. Moreover, the* (N2) *diagrams of definition 7 commute.* ■

We proceed to update and dereferencing maps.

Definition 40 For any type A we define $\ddot{\mathtt{upd}}_A : \mathbb{A}_A \otimes [\![A]\!] \longrightarrow \ddot{T}1$ and $\ddot{\mathtt{drf}}_A : \mathbb{A}_A \longrightarrow \ddot{T}[\![A]\!]$ as follows.



From these obtain:

- $\mathtt{upd}_A : \mathbb{A}_A \otimes [\![A]\!] \xrightarrow{\ddot{\mathtt{upd}}_A} \ddot{T}1 \xrightarrow{\ddot{T}\dot{\eta}} T1$
- $\mathtt{drf}_A : \mathbb{A}_A \xrightarrow{\ddot{\mathtt{drf}}_A} \ddot{T}[\![A]\!] \xrightarrow{\ddot{T}\dot{\eta}} T[\![A]\!]$ ▲

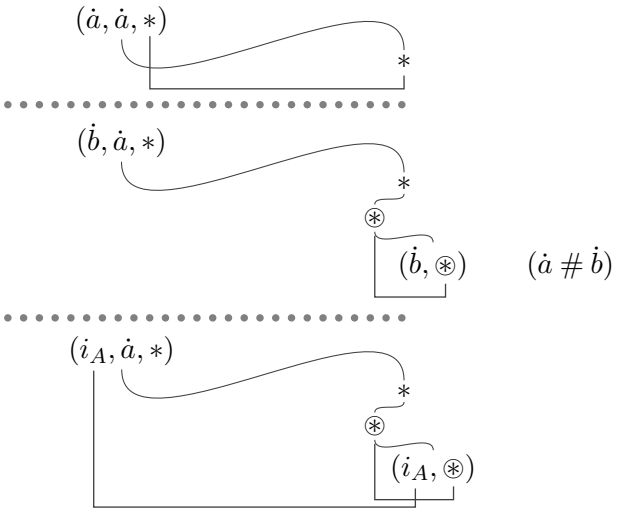
The following proposition is shown by first proving the $\ddot{-}$-versions of the (NR) and (SNR) diagrams and then deriving the original ones.

Proposition 41 *The* (NR,SNR) *diagrams of definition 7 commute for* upd, drf *defined as above.* ■

Finally, we proceed to the exception-related morphisms of the $\lambda_{\nu\epsilon\rho}$-specifications.

Definition 42 For each object A and each name $\dot{a} \in \mathbb{A}_E$, define the morphisms:

- $\mathtt{inx}_A : Q^{\dot{a}}1 \xrightarrow{\mathtt{in}_2} \dot{T}A \xrightarrow{\ddot{\eta}} TA$
- $\mathtt{hdl}_A : TA \otimes Q^{\dot{a}}1 \otimes TA \xrightarrow{\ddot{\tau}'} \ddot{T}(\dot{T}A \otimes Q^{\dot{a}}1 \otimes TA) \xrightarrow{\ddot{T}\dot{\mathtt{hdl}}_A ; \ddot{\mu}} TA$
- $\dot{\mathtt{hdl}}_A : \dot{T}A \otimes Q^{\dot{a}}1 \otimes TA \longrightarrow TA$



▲

Proposition 43 *The above defined arrows make the* (NE) *diagrams commute.* ■

The above constructions are sufficient for a $\lambda_{\nu\epsilon\rho}$-model. For adequacy we need the following property of values and exceptions.

Lemma 44 (Values and Exceptions) *Let $\vec{a} \mid \varnothing \vdash M : A$ be a typed term. For any store S, if $S \models M$ is non-reducing then*

I. *if M is a non-value then for no $\vec{b}, i_A$ do we have $[(\vec{a}, *) * \circledast (i_A, \circledast)^{\vec{b}}] \in [\![\bar{S}\,; M]\!]$,*

II. *if M is a non-exception then for no $\vec{b}, \dot{a}$ do we have* $[(\vec{a}, *) * \circledast (\dot{a}, \circledast)^{\vec{b}}] \in [\![\bar{S}\,; M]\!]$. ■

Proposition 45 ($\mathcal{V}_{\mathtt{t}}$ a model) *$\langle \mathcal{V}_{\mathtt{t}}, T, Q \rangle$ is an adequate $\lambda_{\nu\varepsilon\rho}$-model.*

Proof: We show each point of definition 7.

I. Shown in proposition 29.

II. Shown in proposition 37.

III. Standard.

IV. Straightforward, using constructions from definition 21.

V. Shown in propositions 32,39.

VI. Shown in proposition 41.

VII. Shown in proposition 43.

Finally, adequacy follows from O-adequacy (lemma 56) which is proven using lemma 44. ■

Hence, $\mathcal{V}_{\mathtt{t}}$ is a sound model for $\nu\varepsilon\rho$. The next question to consider is *whether it is fully abstract*, that is whether it satisfies some definability requirement. The answer to this question is negative: in our game semantics we have included store- and exception-related behaviors that are disallowed in the operational semantics.

- Firstly, our strategies treat the store ξ like any other arena, while in the reduction calculus the treatment of store follows some basic guidelines. For example, if a store S is updated to S' then the original store S is not accessible any more. In strategies we do not have such a condition: in a play there may be several ξ's opened, yet there is no discipline on which of these are accessible to Player whenever he makes a move. Another condition involves the fact that a store either 'knows' the value of a name or it doesn't know it. Hence, when a name is asked, the store either returns its value or it deadlocks; there is no third option. In a play, however, when Opponent asks the value of some name, Player is free to evade answering and play somewhere else.

- Moreover, our strategies *may well handle fresh (or unknown) exceptions*, whereas this is not possible in the operational semantics: there, a fresh exception always escapes out of its context.

In order to obtain a fully abstract semantics we will have to constrain strategies.

4.3 A fully abstract semantics in x-tidy strategies

We constrain total strategies be imposing two forms of discipline: one concerning the good use of store (*tidiness*), and one concerning the non-handling of fresh exceptions. For this discipline to be applicable we also need to constrain the available arenas to those that may appear in the domain or codomain of a term's translation. For these arenas we can specify which moves are related to the store and which are exceptions.

Definition 46 Consider $\mathcal{V}_{\nu\epsilon\rho}$, the full subcategory of $\mathcal{V}_{\mathfrak{t}}$ with objects defined as follows.

$$Ob(\mathcal{V}_{\nu\epsilon\rho}) \ni A, B ::= 1 \mid \mathbb{N} \mid \mathbb{A}^{\vec{a}} \mid A \otimes B \mid A \mathrel{\tilde{\Rightarrow}} TB$$

For each such arena A we define its set of ***store-Handles***, H_A, and its set of ***Exception-moves***, X_A, as follows.

$$\begin{array}{ll} H_1 = H_{\mathbb{N}} = H_{\mathbb{A}^{\vec{a}}} \triangleq \varnothing & H_{A\otimes B} \triangleq H_A \cup H_B \\ H_{A \tilde{\Rightarrow} TB} \triangleq \{(i_A, \circledast_A), (i_B, \circledast_B), (\dot{a}, \circledast_B)\} \cup H_A \cup H_B \cup H_{\xi_A} \cup H_{\xi_B} & \\ H_\xi \triangleq \bigcup_C H_{[\![C]\!]} & \text{if } \xi = \bigotimes_C (\mathbb{A}_C \Rightarrow [\![C]\!]) \end{array}$$

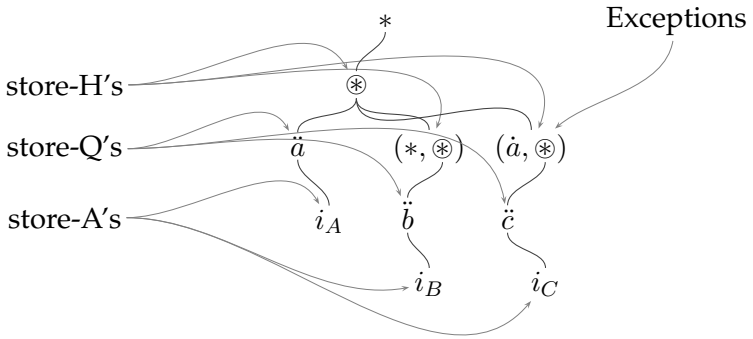
$$\begin{array}{ll} X_1 = X_{\mathbb{N}} = X_{\mathbb{A}^{\vec{a}}} \triangleq \varnothing & X_{A\otimes B} \triangleq X_A \cup X_B \\ X_{A \tilde{\Rightarrow} TB} \triangleq \{(\dot{a}, \circledast_B)\} \cup X_A \cup X_B \cup X_{\xi_A} \cup X_{\xi_B} & \\ X_\xi \triangleq \bigcup_C X_{[\![C]\!]} & \text{if } \xi = \bigotimes_C (\mathbb{A}_C \Rightarrow [\![C]\!]) \end{array}$$

where we write $A \tilde{\Rightarrow} TB$ as $A \tilde{\Rightarrow} (\xi_A \Rightarrow (B + \mathbb{A}_E) \otimes \xi_B)$. In an arena A, a store-handle justifies (all) questions of the form $\ddot{a}$, which we call ***store-Questions***. Answers to store-questions are called ***store-Answers***. ▲

Note in particular that $[\![A]\!], Q^{\vec{a}}[\![A]\!], T[\![A]\!] \in Ob(\mathcal{V}_{\nu\epsilon\rho})$, for each type A, where we assume $T[\![A]\!] = 1 \tilde{\Rightarrow} T[\![A]\!]$. There is a circularity in $H_{A \tilde{\Rightarrow} TB}$ and $X_{A \tilde{\Rightarrow} TB}$ in the above definition; what is hidden is a definition by induction on the level of moves.

Below we give an example of how the above classes of moves are related.

$$T1 = \quad \xi \Rightarrow (1 + \mathbb{A}_E) \otimes \xi$$



From now on we work in $\mathcal{V}_{\nu\varepsilon\rho}$, unless stated otherwise. A first property we can show is that a move is exclusively either initial or a strore-H -Q -A.

Proposition 47 *For any type $A \in Ob(\mathcal{V}_{\nu\varepsilon\rho})$,*

$$M_A = I_A \uplus H_A \uplus \{m \in M_A \mid m \textit{ a store-Q}\} \uplus \{m \in M_A \mid m \textit{ a store-A}\}$$

Proof: By induction on the level of m, $l(m)$, inside A, and on the inductive level of A (resulting from the inductive definition of $Ob(\mathcal{V}_{\nu\varepsilon\rho})$), $|A|$. ■

The notions of store-handles and exception-moves can be straightforwardly extended to prearenas. Around these notions we define x-tidy strategies. Note that we endorse the following notational convention. Since stores ξ may occur in several places inside a (pre)arena we may use parenthesized indices to distinguish identical moves from different stores. For example, the same store-question q may be occasionally denoted $q_{(O)}$ or $q_{(P)}$, the particular notation denoting the OP-polarity of the moves. Moreover, by O-store-H's we mean store-H's played by Opponent, etc., and by X-moves we mean exception-moves.

Definition 48 A total strategy σ is ***x-tidy*** if whenever odd-length $[s] \in \sigma$ then:

(TD1) If s ends in a store-Q q then $[sx] \in \sigma$, with x being either a store-A to q introducing no new names, or a copy of q. In particular, if $q = \ddot{a}^{\vec{a}}$ with $\ddot{a} \# \ulcorner s \urcorner^{-}$ then the latter case holds.

(TD2) If $[sq_{(P)}] \in \sigma$ with q a store-Q then $q_{(P)}$ is justified by last O-store-H in $\ulcorner s \urcorner$.

(TD3) If $\ulcorner s \urcorner = s' q_{(O)} q_{(P)} t\, y_{(O)}$ with q a store-Q then $[s\, y_{(P)}] \in \sigma$ with $y_{(P)}$ justified by $\ulcorner s \urcorner$.-3.

(xTD1) If s ends in an X-move $(\dot{a}, \circledast)^{\vec{a}}$ with $\dot{a} \# \ulcorner s \urcorner^{-}$ then $[s(\dot{a}, \circledast)^{\vec{a}'}] \in \sigma$.

(xTD3) If $\ulcorner s \urcorner = s'(\dot{a}, \circledast)^{\vec{a}}_{(O)} (\dot{a}, \circledast)^{\vec{a}}_{(P)} q_{(O)}$ with q a store-Q, $(\dot{a}, \circledast)_{(O)}$ an X-move and $\dot{a} \# s'$ then $[s\, q_{(P)}] \in \sigma$. ▲

The (TD) conditions define tidy strategies of [17] and impose a certain store-discipline: *when a store-Q is encountered Player either answers with a value or copycats the store-Q to the previous store-H and continues to copycat thereon; the latter happens especially when the store-Q is a fresh one*. The (xTD) conditions provide a fresh-exception-discipline: *when a fresh exception is encountered Player must copycat it*. (xTD3) in particular states that no store-updates occur when forwarding a fresh exception.

In the definition of x-tidiness, $(\dot{a}, \circledast)^{\vec{a}'}_{(P)}$ is an answer and hence needs to be justified by the pending question; the following lemma shows that this is always possible.

Lemma 49 *If odd-length $[s] \in \sigma$ ends in an X-move $(\dot{a}, \circledast)^{\vec{a}}$ then s has a pending-Q which is an O-store-H, and $s(\dot{a}, \circledast)^{\vec{a}}$ is a play.*

Proof: s being odd-length implies that it has a pending question, say q. If q were a P-move then $s = s_1 q s_2$ with s_1, s_2 being odd-length, so an A in s_2 should be justified by q, $\lightning$. Hence, q an O-move. Moreover, q cannot be initial, by totality, and neither a store-Q: q being unanswered would mean that P copycats after it, so the move following q would be a copy of it answered by an O-store-A y, say. After y is played, P must answer q with a copy of y, thus y can only be the last move in s, i.e. $(\dot{a}, \circledast)^{\vec{a}'}$, $\lightning$ as y a store-A. Hence, q an O-store-H. Thus, $s(\dot{a}, \circledast)^{\vec{a}'}$ satisfies well-bracketing, and it clearly satisfies NC's. Finally, it also satisfies visibility since s and $\ulcorner s \urcorner$ have the same pending-Q. ■

By close inspection on how (TD) and (xTD) conditions interact with play-composition, we can show the following.

Proposition 50 *If $\sigma : A \longrightarrow B$ and $\tau : B \longrightarrow C$ are x-tidy strategies then so is $\sigma ; \tau$.* ■

Moreover, it is straightforward to see that identity strategies are x-tidy. Hence, we can define our category of x-tidy strategies.

Definition 51 ($\mathcal{T}$) $\mathcal{xT}$ is the lluf subcategory of $\mathcal{V}_{\nu\varepsilon\rho}$ of x-tidy strategies. Moreover, $\mathcal{xT}_{\mathtt{tt}}$ is the lluf subcategory of $\mathcal{xT}$ containing ttotal strategies. ▲

In fact, strategies in general that mainly do copycats are easily shown to be x-tidy. With this observation it is not difficult to show the following.

Proposition 52 ($\mathcal{V}_{\mathtt{t}}$ to $\mathcal{xT}$) *$\mathcal{xT}$ forms an adequate $\lambda_{\nu\varepsilon\rho}$-model by inheriting the necessary structure from $\mathcal{V}_{\mathtt{t}}$, i.e. for all $\vec{a}$,*

1. *If $f : A \to B$, $g : A \to C$ are x-tidy then $\langle f, g \rangle$ is. Moreover, projections and terminal arrows are all x-tidy.*
2. *$\eta_A, \mu_A, \tau_{A,B}, \theta_A$ are all x-tidy, and if h is x-tidy then Th is. Moreover, $f : A \otimes B \to TC$ is x-tidy iff $\Lambda^T(f)$ is.*
3. *Successor, predecessor and natural number arrows are x-tidy.*
4. *Name-equality arrows for references are x-tidy.*
5. *ε_A, δ_A are x-tidy, and if h is x-tidy then so is $Q^{\vec{a}}h$. Moreover, $(\frac{\vec{a}}{\vec{a}'})_A$ and $\mathtt{new}_A^{\vec{a}a}$ are x-tidy.*
6. *$\mathtt{upd}_A, \mathtt{drf}_A$ are x-tidy.*
7. *$\mathtt{inx}_A, \mathtt{hdl}_A$ are x-tidy.* ■

Henceforth, by strategies we shall mean x-tidy strategies, unless stated otherwise. We proceed to show p-observationality for $\mathcal{xT}$. We define the following observability predicate and the related semantic preorder.

Definition 53 ($O, \lesssim$) Expand $\mathcal{xT}$ to $\langle \mathcal{xT}, T, Q, O, \lesssim \rangle$ by setting, for each $\vec{a}$,

- $O^{\vec{a}} \triangleq \{f \in \mathcal{K}\mathcal{T}(Q^{\vec{a}}1, T\mathbb{N}) \mid \exists \vec{b}.\, [(\vec{a}, *) * \circledast (0, \circledast)^{\vec{b}}] \in f\}$
- for each $f, g \in \mathcal{K}\mathcal{T}(Q^{\vec{a}}A, TB)$, $f \lesssim^{\vec{a}} g$ if

$$\forall \rho \in \mathcal{K}\mathcal{T}(Q^{\vec{a}}(A \mathrel{\tilde{\Rightarrow}} TB), T\mathbb{N}).\, (\Lambda^{Q^{\vec{a}}}(f)\,;\rho \in O^{\vec{a}} \Longrightarrow \Lambda^{Q^{\vec{a}}}(g)\,;\rho \in O^{\vec{a}})$$

  with notation as in definition 12. ▲

We find it useful to restrict the class of test-arrows ρ that participate in the definition of the intrinsic preorder. A total arrow $f : A \to B$ is ***tl4*** if for any $[i_A\, i_B\, j_B] \in f$ there exists $[i_A\, i_B\, j_B\, j_A^{\vec{a}}] \in f$, and whenever $[s\, x\, j_A'^{\vec{a}'}] \in f$ then $x \in J_B$.

Lemma 54 (tl4 tests suffice) *For each $\vec{a}$ and $f, g \in \mathcal{K}\mathcal{T}(Q^{\vec{a}}A, TB)$, $f \lesssim^{\vec{a}} g$ iff*

$$\forall \rho \in \mathcal{K}\mathcal{T}(Q^{\vec{a}}(A \mathrel{\tilde{\Rightarrow}} TB), T\mathbb{N}).\, \rho \text{ is tl4} \implies (\Lambda^{Q^{\vec{a}}}(f)\,;\rho \in O^{\vec{a}} \implies \Lambda^{Q^{\vec{a}}}(g)\,;\rho \in O^{\vec{a}})$$

■

It is not difficult now to show the following.

Lemma 55 *For any morphism $f : Q^{\vec{a}a}1 \to B$, with B pointed, and any tl4 morphism $\rho : Q^{\vec{a}}B \to T\mathbb{N}$,*

$$\delta\,;Q^{\vec{a}}\langle a\rangle f\,;\rho \in O^{\vec{a}} \iff \delta\,;Q^{\vec{a}a}f\,;\frac{\vec{a}a}{\vec{a}}\,;\rho \in O^{\vec{a}a}$$

Moreover, for each $\vec{a}a$ and relevant f, g,

$$f \lesssim^{\vec{a}a} g \implies \langle a\rangle f \lesssim^{\vec{a}} \langle a\rangle g$$
$$f \lesssim^{\vec{a}} g \implies \frac{\vec{a}a}{\vec{a}}\,;f \lesssim^{\vec{a}a} \frac{\vec{a}a}{\vec{a}}\,;g$$

■

We proceed in proving O-adequacy, which will be crucial for showing adequacy and p-observationality.

Lemma 56 (O-Adequacy) *Let $\vec{a} \mid \varnothing \vdash M : \mathbb{N}$ be a typed term. If $[\![M]\!] \in O^{\vec{a}}$ then there exists some S such that $\vec{a} \Vdash M \longrightarrow\!\!\!\!\!\rightarrow S \Vdash 0$.*

Proof: By lemma 44 it suffices to show that, for any such M, there is a non-reducing sequent $S \Vdash N$ such that $\vec{a} \Vdash M \longrightarrow\!\!\!\!\!\rightarrow S \Vdash N$, as then N would be a closed value of type $\mathbb{N}$ such that $[\![\bar{S}\,;N]\!] \in O^{\vec{a}}$ –and therefore $N = 0$. But then it suffices to show that there is no infinite reduction sequence starting from $\vec{a} \Vdash M$ and containing infinitely many DRF reduction steps: leaving DRF's aside we are left with a ν-calculus with some non-recursive effects, and the ν-calculus is strongly normalising for closed terms (v. [16]). To show this we will use an operation on terms adding new-name constructors just before dereferencings. The operation yields, for each term M, a term $(M)^\circ$ the semantics of which is equivalent to that of M. On the other hand, $\vec{a} \Vdash (M)^\circ$ cannot perform infinitely many DRF reduction steps without creating infinitely many new names.

So, for each term M define $(M)^\circ$ as follows.

$$(!N)^\circ \triangleq (\nu a.a); !(N)^\circ\,,\ (x)^\circ \triangleq x\,,\ (\lambda x.M)^\circ \triangleq \lambda x.(M)^\circ\,,\ etc.$$

We show that $[\![(M)^\circ]\!] \backsimeq [\![M]\!]$, by induction on M; the base cases are trivial. The induction step follows immediately from the IH and the fact that $\backsimeq$ is a congruence, in all cases except for M being $!N$. In the latter case we have that $[\![(M)^\circ]\!] = \mathtt{new}\,; T\frac{\vec{a}a}{\vec{a}}\,; T[\![!(N)^\circ]\!]\,; \mu$, while the IH implies that $[\![M]\!] \backsimeq [\![!(N)^\circ]\!]$. By properties of $\lesssim$ we have that, for any $f : Q^{\vec{a}}A \to TB,\ f \backsimeq \langle a\rangle(\frac{\vec{a}a}{\vec{a}}\,; f)\ (= \mathtt{new}\,; T\frac{\vec{a}a}{\vec{a}}\,; Tf\,; \mu)$, and hence $[\![(M)^\circ]\!] \backsimeq [\![M]\!]$.

Now take any $\vec{a} \mid \varnothing \vdash M : \mathbb{N}$ and assume $[\![M]\!] \in O^{\vec{a}}$, and that $\vec{a} \models M$ diverges using infinitely many DRF reduction steps. Then, $\vec{a} \models (M)^\circ$ diverges using infinitely many NEW reduction steps. However, since $[\![(M)^\circ]\!] \backsimeq [\![M]\!]$, we have $[\![(M)^\circ]\!] \in O^{\vec{a}}$ and therefore $[(\vec{a}, *) * \circledast (0, \circledast)^{\vec{b}}] \in [\![(M)^\circ]\!]$ for some $\vec{b}$. However, $\vec{a} \models (M)^\circ$ reduces to some $S \models M'$ using $|\vec{b}| + 1$ NEW reduction steps, so, by correctness, $[\![(M)^\circ]\!] = \langle\!\langle\vec{c}\rangle\!\rangle[\![\bar{S}\,; M']\!]$ with $|\vec{c}| = |\vec{b}| + 1$, $\lightning$. ■

We can now show p-observationality.

Proposition 57 (p-Observationality) *$\mathcal{xT}$ is p-observational.*

Proof: We only need to show that, for any $\vec{a} \mid \varnothing \vdash M : \mathbb{N}$, $[\![M]\!] \in O^{\vec{a}}$ iff $\exists S, \vec{b}.\, [\![M]\!] = \langle\!\langle\vec{b}\rangle\!\rangle[\![\bar{S}\,; 0]\!]$. The "if"-part is straightforward. Conversely, let $[\![M]\!] \in O^{\vec{a}}$. By O-adequacy $\vec{a} \models M$ converges to some $S \models 0$. Hence, $[\![M]\!] = \langle\!\langle\vec{b}\rangle\!\rangle[\![\bar{S}\,; 0]\!]$. ■

We are left to show p-definability. For this we need to isolate a class of definable arrows which suffice for defining the intrinsic preorder. We say a strategy σ is ***finitary*** if $\mathtt{trunc}(\sigma)$ is finite, where $\mathtt{trunc}(\sigma)$ is the subset of $\mathtt{viewf}(\sigma)$ which excludes all the store-copycats, all default initial answers –the latter dictated by totality– and all fresh-exception copycats.

Theorem 58 (Definability) *Let A, B be types and $\sigma : Q^{\vec{a}}[\![A]\!] \longrightarrow T[\![B]\!]$ be finitary. Then σ is definable.*

Proof outline: The proof proceeds by first showing a Decomposition Lemma, which states that we can decompose a strategy $\sigma : Q^{\vec{a}}[\![A]\!] \longrightarrow T[\![B]\!]$ as follows.

1. If σ is non-empty then we can decompose it to a (possibly infinite) set of strategies, taking one strategy for each distinct initial move $[(\vec{a}, i_A)]$ for which σ has a response $[(\vec{a}, i_A) * \circledast x]$.

2. If σ has a response $[(\vec{a}, i_A) * \circledast x]$ to exactly one initial move $[(\vec{a}, i_A)]$ then we can express it as an abstraction of a 4th-move-non-introducing strategy σ'.

3. If σ has a response $[(\vec{a}, i_A) * \circledast x]$ to exactly one initial move $[(\vec{a}, i_A)]$ and that response is non-introducing then,

(a) if x is a store-Q $\ddot{a}$ then σ can be expressed as a composition of $[\![!\ddot{a}]\!]$ with a strategy σ',

(b) otherwise, for each $\ddot{a}$ such that σ answers $[(\vec{a}, i_A) * \circledast x\, \ddot{a}]$, σ can be expressed as a composition of a store-update of $\ddot{a}$ and of a strategy $\sigma' = \sigma -$(update of $\ddot{a}$).

The proof of definability is then done by induction on $d(\sigma) \triangleq (|\mathtt{trunc}(\sigma)|, \|\sigma\|)$, where $\|\sigma\|$ measures how many names are introduced in maximum plays of σ. In the above decomposition d decreases (note that σ being finitary implies that the above decomposition is finite), hence the problem of definability of σ is reduced to that of definability of a finitary strategy σ_0 of equal length, but with σ_0 having no initial effects (i.e. fresh-name creation, name-update or name-dereferencing). On σ_0 we then apply almost verbatim the methodology of [7]. ■

Hence, setting

$$D^{\vec{a}}_{A,B} \triangleq \{f : Q^{\vec{a}}[\![A]\!] \longrightarrow T[\![B]\!] \mid f \text{ is finitary}\}$$

and following a more or less standard method we can obtain p-definability, and thus full abstraction.

Theorem 59 *$\chi\mathcal{T}$ is a fully abstract model of $\nu\in\rho$.* ■

References

[1] ABRAMSKY, S., GHICA, D., MURAWSKI, A., ONG, L., AND STARK, I. Nominal games and full abstraction for the nu-calculus. In *Proceedings of LICS '04* (2004).

[2] ABRAMSKY, S., JAGADEESAN, R., AND MALACARIA, P. Full abstraction for PCF. *Information and Computation 163*, 2 (2000).

[3] BECK, J. Distributive laws. In *Seminar on Triples and Categorical Homology Theory, Zürich, 1966/67*, vol. 80 of *LNM*. 1969, pp. 119–140.

[4] BROOKES, S., AND GEVA, S. Computational comonads and intensional semantics. In *Applications of Categories in Computer Science: Proc. LMS Symp., July 1991*. 1992, pp. 1–44.

[5] BROOKES, S., AND VAN STONE, K. Monads and comonads in intensional semantics. Tech. Rep. CMU-CS-93-140, Pittsburgh, PA, USA, 1993.

[6] GABBAY, M. J., AND PITTS, A. M. A new approach to abstract syntax with variable binding. *Formal Aspects of Computing 13* (2002), 341–363.

[7] HONDA, K., AND YOSHIDA, N. Game-theoretic analysis of call-by-value computation. *Theoretical Computer Science 221*, 1–2 (1999), 393–456.

[8] HYLAND, J. M. E., AND ONG, C.-H. L. On full abstraction for PCF: I, II, III. *Information and Computation 163*, 2 (2000), 285–408.

[9] KRIVINE, J.-L. *Lambda-calcul, types et modèles*. Masson, 1990.

[10] LAIRD, J. A fully abstract games semantics of local exceptions. In *Proceedings of LICS '01*.

[11] LAIRD, J. A game semantics of names and pointers. To appear in Annals of Pure and Applied Logic.

[12] MCCUSKER, G. *Games and Full Abstraction for a Functional Metalanguage with Recursive Types*. Distinguished Dissertations. Springer-Verlag, London, 1998.

[13] MOGGI, E. Computational lambda-calculus and monads. In *Proc. of LICS '89* (1989), pp. 14–23.

[14] ONG, L. Observational equivalence of third-order Idealized Algol is decidable. In *Proceedings of LICS '02* (2002), pp. 245–256.

[15] PITTS, A. M. Nominal logic, a first order theory of names and binding. *Information and Computation 186* (2003), 165–193.

[16] PITTS, A. M., AND STARK, I. D. B. Observable properties of higher order functions that dynamically create local names, or: What's new? In *Proceedings of MFCS '93* (1993), pp. 122–141.

[17] TZEVELEKOS, N. Full abstraction for nominal general references. In *Proceedings of LICS '07* (2007), pp. 399–410.